



Universidad
Carlos III de Madrid

Departamento de Informática
Ingeniería en Informática

TRABAJO DE FIN DE GRADO

Aplicación móvil basada en redes
sociales para dar soporte a la
compartición de material docente
(Plataforma Android)

Autor: Santiago García Delgado

Tutor: Javier García Guzmán

Leganés, Febrero de 2013

Agradecimientos

Es complejo agradecer en unas pocas líneas el apoyo y cariño que tanta y tanta gente me ha dado a lo largo de estos cuatro años y medio, por lo que nombrar a todos y cada uno de ellos sería físicamente imposible, por ello pido perdón, y pese a que no figuréis en el “*hall of fame*” de este trabajo vuestra labor no ha sido menos importante que la de otros.

En primer lugar me gustaría agradecer la figura de todos y cada uno de los profesores que he tenido el gusto de conocer en la carrera, desde el primero hasta el último, ya que todos, de una u otra manera, han ayudado a mi formación académica, y en muchos aspectos personal.

En segundo lugar me gustaría agradecerle a mi tutor, Javier García Guzmán, su apoyo y dedicación, gracias a lo cual este trabajo ha llegado a buen puerto. También me gustaría agradecerle a Javier Sánchez todo el apoyo prestado en el desarrollo del trabajo, que objetivamente no ha sido poco. Por último me gustaría agradecer a todos los chicos del SEL su amabilidad al acogerme con los brazos abiertos en su despacho.

En tercer lugar me gustaría agradecer todo el apoyo, y la paciencia que han tenido conmigo en multitud de ocasiones, los miembros de mi familia, entre los que se encuentran mi hermano Javier y mis padres Milagros y Antonio, además de mi abuela Josefa y mi novia Miriam. Todos ellos han tenido que lidiar con estados de humor irritantes por mi parte, producto de la presión que en muchos casos el trabajo y exigencias de la Universidad ejercían sobre mí. Gracias.

En cuarto lugar me gustaría agradecer a mis compañeros de clase, y más concretamente a mi núcleo duro dentro clase, su apoyo y compañerismo, y todas las horas de compañía que hemos compartido en los días previos a las entregas de prácticas en las aulas *Linux*. Gracias especialmente a Marta, mi compañera de prácticas durante la mayor parte de la carrera, por su apoyo sin límites y su gran dedicación. Has sido, sin duda alguna, la mejor compañera de prácticas que hubiera podido tener.

Para concluir este pequeño rincón para el recuerdo me gustaría mencionar a mis amigos, los cuales me han ayudado cada uno a su manera, pero con la mejor intención del mundo. Gracias a todos ellos he conseguido abstraerme de las obligaciones de la Universidad, aunque fuera por un ratito, hasta en los momentos de mayor tensión. Gracias.

*“No podemos cambiar las cartas que se nos reparten,
solo podemos elegir como jugar la mano”*

Randy Pausch

Resumen

Las plataformas de aprendizaje en línea ofrecen al usuario la posibilidad de acceder a material académico de casi cualquier tipo sin importar su localización, horarios o fechas, facilitando de este modo el aprendizaje de los alumnos mediante a su gran flexibilidad.

El problema que resuelve este trabajo es la dependencia existente en las plataformas de aprendizaje actuales con respecto a los sistemas de donde extraen la información, creando una alternativa a esta ideología para la distribución de material docente entre profesores y alumnos. Junto con la importancia adquirida en los últimos años por los teléfonos inteligentes y las tabletas, todo en conjunto ha motivado la elaboración de este Trabajo de Fin de Grado.

El objetivo principal de este trabajo es desarrollar una aplicación cliente que permita la compartición de material docente entre profesores y alumnos, independiente de la plataforma, mediante el uso de tabletas.

La metodología usada para el desarrollo del trabajo ha sido una metodología ágil de desarrollo, concretamente Scrum. Se ha escogido esta opción porque sus características se adaptaban a la planificación adaptativa e incremental que hemos seguido en todo el proceso de desarrollo del trabajo.

La plataforma sobre la que se ha desarrollado el trabajo ha sido Android, y se han utilizado las herramientas de desarrollo comunes para este tipo de trabajos, entre las que se encuentra como entorno para el desarrollo Eclipse, y como lenguaje de programación Java.

El resultado del trabajo ha sido una aplicación móvil que permite a los profesores la compartición de recursos de una manera sencilla con sus alumnos a través de dispositivos móviles como son las tabletas.

Índice general

CAPÍTULO 1: INTRODUCCIÓN

1.1 AREA DE TRABAJO.....	2
1.2 PROBLEMA	2
1.3 OBJETIVOS	3
1.4 MÉTODO DE TRABAJO	4
1.5.1 GLOSARIO DE TÉRMINOS.....	4
1.5.2 ABREVIATURAS	5
1.6 CONTENIDO DE LA MEMORIA.....	6

CAPÍTULO 2: ESTADO DEL ARTE

2.1 ANÁLISIS DE APLICACIONES WEB Y MÓVIL RELACIONADAS CON LOS OBJETIVOS ESTABLECIDOS	10
2.1.1 WC3SCHOOLS	10
2.1.2 BUSUU	11
2.1.3 BLACKBOARD	12
2.1.4 KHAN ACADEMY	13
2.1.5 WIKILEARNING	14
2.1.6 SLIDESHARE	15
2.2 ENTORNO DE DESARROLLO MÓVIL SELECCIONADO	17
2.2.1 ENTORNOS MÓVILES.....	17
2.2.2 VISIÓN GENERAL SOBRE ANDROID	18
2.2.3 ESTRUCTURA DE ANDROID	20
2.2.4 ENTORNO DE DESARROLLO INTEGRADO.....	21
2.3 DESARROLLO ÁGIL COMO MÉTODO PARA REALIZAR EL PROYECTO.....	22
2.3.1 THE AGILE ALLIANCE	22
2.3.2 SCRUM.....	22
2.4 CONCLUSIONES DEL ANÁLISIS	24

CAPÍTULO 3: ESPECIFICACIÓN DE REQUISITOS

3.1 INTRODUCCIÓN	27
3.2 DESCRIPCIÓN GENERAL DEL SISTEMA A DESARROLLAR	27
3.2.1 PERSPECTIVAS DEL PRODUCTO	27
3.2.2 FUNCIONES DEL PRODUCTO	28
3.2.3 CARACTERÍSTICAS DE LOS USUARIOS	29
3.2.4 INTERACCIÓN CON OTROS SISTEMAS	29
3.2.5 RESTRICCIONES	29
3.2.6 SUPOSICIONES Y DEPENDENCIAS	30
3.3 REQUISITOS ESPECÍFICOS	30
3.3.1 REQUISITOS FUNCIONALES	32
3.3.2. INTERFACES EXTERNAS	36
3.3.3 REQUISITOS DE RENDIMIENTO	42
3.3.4 RESTRICCIONES DE DISEÑO	42
3.3.5 ATRIBUTOS DEL SISTEMA	44

CAPÍTULO 4: DISEÑO DE LA APLICACIÓN

4.1 PROTOTIPO	47
4.1.1 MOLE ALUMNOS	47
4.1.2 MOLE PROFESORES	52
4.2 ALTERNATIVAS DE DISEÑO	61
4.2.1 DISEÑO DE LA APLICACIÓN	64
4.3 DIAGRAMA DE CLASES	64
4.3.1 DIAGRAMA DE CLASES MOLE ALUMNOS	65
4.3.2 DIAGRAMA DE CLASES MOLE PROFESORES	68
4.4 DIAGRAMAS DE SECUENCIA	74

CAPÍTULO 5: PRUEBAS DE LA APLICACIÓN

5.1 PRUEBAS UNITARIAS	89
5.2 PRUEBAS DE INTEGRACIÓN	91

5.3 PRUEBAS DE SISTEMA	92
CAPÍTULO 6: CONCLUSIONES Y LÍNEAS FUTURAS	
6.1 CONCLUSIONES	104
6.2 LÍNEAS FUTURAS	105

Lista de figuras

Figura 2.1.1: Wc3schools.....	11
Figura 2.1.2: Busuu	12
Figura 2.1.3: Blackboard.....	13
Figura 2.1.4: Khan academy	14
Figura 2.1.5: Wikilearning	14
Figura 2.1.6: Slideshare.....	15
Figura 2.2.1.1: Terminales operativos 2012.....	17
Figura 2.2.1.2: Ventas terminales 2012.....	18
Figura 2.2.2: Versiones Android.....	19
Figura 2.2.3: Estructura Android.....	20
Figura 2.3.2: Scrum.....	23
Figura 4.1.1.1: Vista inicio.....	48
Figura 4.1.1.2: Vista principal.....	48
Figura 4.1.1.3: Suscripción curso.....	49
Figura 4.1.1.4: Baja curso	50
Figura 4.1.1.5: Baja emergente	50
Figura 4.1.1.6: Tablón alumno	51
Figura 4.1.2.1: Vista inicio.....	52
Figura 4.1.2.2: Vista principal.....	53
Figura 4.1.2.3: Vista alumnos	53
Figura 4.1.2.4: Borrado alumnos.....	54
Figura 4.1.2.5: Borrado alumno emergente.....	55
Figura 4.1.2.6: Modificar curso.....	56
Figura 4.1.2.7: Modificar curso emergente	56
Figura 4.1.2.8: Formulario modificación	57
Figura 4.1.2.9: Formulario modificación adjuntar	58

Figura 4.1.2.10: Explorador archivos local	58
Figura 4.1.2.11: Explorador archivos remoto.....	59
Figura 4.1.2.12: Añadir nueva entrada.....	60
Figura 4.1.2.13: Añadir entrada a sesión.....	60
Figura 4.1.2.14: Añadir sesión nueva.....	61
Figura 4.2.1: Diseño de la aplicación.....	64
Figura 4.3.1.1: diagrama 1	65
Figura 4.3.1.2: diagrama 2	66
Figura 4.3.1.3: diagrama 3	66
Figura 4.3.1.4: diagrama 4	67
Figura 4.3.1.5: diagrama 5	68
Figura 4.3.2.1: diagrama 1	69
Figura 4.3.2.2: diagrama 2	70
Figura 4.3.2.3: diagrama 3	71
Figura 4.3.2.4: diagrama 4	72
Figura 4.3.2.5: diagrama 5	73
Figura 4.3.2.6: diagrama 6	74
Figura 4.4.1: diagrama 1	75
Figura 4.4.2: diagrama 2	76
Figura 4.4.3: diagrama 3	77
Figura 4.4.4: diagrama 4	78
Figura 4.4.5: diagrama 5	79
Figura 4.4.6: diagrama 6	79
Figura 4.4.7: diagrama 7	80
Figura 4.4.8: diagrama 8	82
Figura 4.4.9: diagrama 9	83
Figura 4.4.10: diagrama 10	84

Figura 4.4.11: diagrama 11	86
----------------------------------	----

Lista de tablas

Tabla 2.1: Comparativa aplicaciones	16
Tabla 2.2.2: Versiones Android	19
Tabla 3.3.1: Modelo	31
Tabla 3.3.1.1: Requisito funcional 1	32
Tabla 3.3.1.2: Requisito funcional 2	32
Tabla 3.3.1.3: Requisito funcional 3	32
Tabla 3.3.1.4: Requisito funcional 4	33
Tabla 3.3.1.5: Requisito funcional 5	33
Tabla 3.3.1.6: Requisito funcional 6	33
Tabla 3.3.1.7: Requisito funcional 7	34
Tabla 3.3.1.8: Requisito funcional 8	34
Tabla 3.3.1.9: Requisito funcional 9	34
Tabla 3.3.1.10: Requisito funcional 10	35
Tabla 3.3.1.11: Requisito funcional 11	35
Tabla 3.3.1.12: Requisito funcional 12	35
Tabla 3.3.1.13: Requisito funcional 13	36
Tabla 3.3.2.1: Requisito de interfaz 1	36
Tabla 3.3.2.2: Requisito de interfaz 2	37
Tabla 3.3.2.3: Requisito de interfaz 3	37
Tabla 3.3.2.4: Requisito de interfaz 4	37
Tabla 3.3.2.5: Requisito de interfaz 5	38
Tabla 3.3.2.6: Requisito de interfaz 6	38
Tabla 3.3.2.7: Requisito de interfaz 7	38
Tabla 3.3.2.8: Requisito de interfaz 8	39
Tabla 3.3.2.9: Requisito de interfaz 9	39
Tabla 3.3.2.10: Requisito de interfaz 10	39

Tabla 3.3.2.11: Requisito de interfaz 11	40
Tabla 3.3.2.12: Requisito de interfaz 12	40
Tabla 3.3.2.13: Requisito de interfaz 13	40
Tabla 3.3.2.14: Requisito de interfaz 14	41
Tabla 3.3.2.15: Requisito de interfaz 15	41
Tabla 3.3.3.1: Requisito de rendimiento 1	42
Tabla 3.3.4.1: Requisito de diseño 1	42
Tabla 3.3.4.2: Requisito de diseño 2	43
Tabla 3.3.4.3: Requisito de diseño 3	43
Tabla 3.3.4.4: Requisito de diseño 4	43
Tabla 5.1.1: Pruebas unitarias 1	90
Tabla 5.1.2: Pruebas unitarias 2	91
Tabla 5.3.1: Plantilla 1	92
Tabla 5.3.2: Prueba sistema 1.....	93
Tabla 5.3.4: Prueba sistema 2.....	93
Tabla 5.3.4: Prueba sistema 3.....	94
Tabla 5.3.5: Prueba sistema 4.....	94
Tabla 5.3.6: Prueba sistema 5.....	94
Tabla 5.3.7: Prueba sistema 6.....	96
Tabla 5.3.8: Prueba sistema 7.....	97
Tabla 5.3.9: Prueba sistema 8.....	97
Tabla 5.3.10: Prueba sistema 9.....	98
Tabla 5.3.11: Prueba sistema 10.....	99
Tabla 5.3.12: Prueba sistema 11.....	100
Tabla 5.3.13: Prueba sistema 12.....	101
Tabla 6.1: Gastos 1.....	101
Tabla 6.2: Gastos 2.....	101

CAPÍTULO 1 : INTRODUCCIÓN

1.1 AREA DE TRABAJO.....	2
1.2 PROBLEMA	2
1.3 OBJETIVOS	3
1.4 MÉTODO DE TRABAJO	4
1.5.1 GLOSARIO DE TÉRMINOS.....	4
1.5.2 ABREVIATURAS	5
1.6 CONTENIDO DE LA MEMORIA.....	6

El objetivo de este capítulo es realizar una breve introducción al presente trabajo, explicando el área sobre la que se ha operado, el problema que se deseaba resolver, sus motivaciones, objetivos, y el método de desarrollo usado para la elaboración del mismo. Además, y para concluir este capítulo se facilitará un glosario de términos para ayudar a la comprensión del documento, y por ultimo se expondrá la estructura del resto de capítulos contenidos en este trabajo.

1.1 AREA DE TRABAJO

El área en la que se desarrolla este trabajo es la perteneciente al aprendizaje en línea mediante entornos de aprendizaje personal. Un Entorno personal de aprendizaje [1] o *PLE* (*Personal Learning Enviroment*) es un sistema interactivo que permite a un usuario formarse académicamente de forma autónoma e informal haciendo uso de los recursos compartidos en el mismo por otros usuarios. Este tipo de sistemas además de proveer al usuario de material con el que formarse le ofrecen la posibilidad de marcarse sus propias metas personales, gestionar libremente el material que se ofrece para la formación y la comunicación con otros usuarios que también intervienen en el proceso formativo.

Todo *PLE* esta asociado a un Sistema de Gestión de Aprendizaje [2] o *LMS* (*Learning Management System*) que se encarga de gestionar todos los contenidos que se comparten en el *PLE*. Dicho software se localiza en el servidor que da soporte físico a la información del *PLE*.

El primer problema encontrado en este tipo de sistemas [3] es la multitud de ellos que actualmente existen y la poca, o ninguna, relación entre ellos, que provoca que cada uno posea un banco de recursos independiente a los demás. De cara al usuario este hecho se transforma en un impedimento por la duplicidad de documentación comparando los diferentes sistemas de este tipo. Además el usuario es sometido a diferentes políticas de acceso, en muchos casos una por cada *PLE*.

1.2 PROBLEMA

Por cada nueva plataforma que aparece el usuario debe, si quiere poder interactuar con ella, crearse un nuevo perfil en la misma, y siempre que quiera acceder a sus servicios utilizarlo. Debido a la proliferación vertiginosa de entornos con carácter social, y al más que preocupante predominio de la cantidad de los mismos frente a la calidad que ofrecen al usuario, actualmente cada usuario debe gestionar un número ingente de usuarios y contraseñas diferentes para poder acceder a los contenidos de cada plataforma. Por esta razón para un usuario sobresaturado por múltiples perfiles cada vez le resulta más costo y menos necesario acceder a un nuevo sistema que le ofrezca algo parecido a los que ya tiene, y además obligado a efectuar un nuevo registro.

A raíz de este problema se llegó a la conclusión que era necesario crear una nueva aplicación, en forma de cliente, que ofreciera algo que las anteriores no ofrecían,

versatilidad, utilizando identidades sociales de otras plataformas para facilitar al usuario su unión y uso, sin tener que crear un nuevo perfil.

Realizando un estudio de los diferentes sistemas interactivos que actualmente existen, y su popularidad a nivel mundial se planteó que las opciones más viables eran utilizar *Facebook* [4] o *Twitter* [5] como plataforma anfitriona para el uso de nuestra nueva plataforma interactiva. En el año 2012 *Twitter* poseía 383 millones de usuarios a nivel mundial, frente a los más de 1000 millones que *Facebook* poseía en las mismas fechas, por lo que *Facebook* aparecía como la plataforma más atractiva en un primer momento. Tras analizar la funcionalidad inherente a cada una de ellas comprendimos que *Twitter*, aunque poseía menos volumen de usuarios, tenía una funcionalidad más útil de cara a los servicios que nosotros queríamos ofrecer. Por lo tanto *Twitter* fue escogida como plataforma anfitriona para el desarrollo del trabajo.

Otro problema existente en los sistemas de aprendizaje que actualmente existen es su independencia unos de otros, ofreciendo en muchos casos contenidos similares al usuario, pero alojados en diferentes *LMS* que son gestionados únicamente por su *PLE*. Dicha división provocaba al usuario una necesidad de conectarse a un *PLE* para obtener un recurso o a otro para otro diferente, en muchos casos ambos dos complementarios, pero alojados en sitios independientes.

Por esta razón se ha decidido diseñar un *PLE* diferente, que no esté vinculado a un *LMS* único para obtener su información, sino que puede utilizar cualquiera de los existentes con independencia de la plataforma sobre la que se use la aplicación.

1.3 OBJETIVOS

El objetivo principal de este proyecto es crear una parte de un *PLE* que permita a los alumnos de estudios de carácter oficial seguir sus clases matriculadas de forma virtual, y de una forma complementaria, a las plataformas ya existentes. Así mismo existen objetivos secundarios que serán descritos brevemente a continuación:

- **Crear una parte de un PLE que utilice la identidad social del usuario de otras plataformas.** Se utilizarán las cuentas de *Twitter* activas para acceder a nuestra plataforma de aprendizaje, así como para la gestión de asignaturas y alumnos con el fin de facilitar al usuario su acceso.
- **Crear una parte de un PLE con un LMS variable.** Se utilizará un servidor FTP para el almacenamiento de la información, siendo posible, para desarrollos futuros, cambiarlo por un *LMS* de cualquier otro *PLE* ya existente en base a las necesidades.

1.4 MÉTODO DE TRABAJO

El método llevado a cabo para la resolución del problema ha sido iterativo, ya que en un principio no se tenían claros los requisitos que debía cumplir el sistema y se pretendía definirlos a lo largo del desarrollo del proyecto, dependiendo de los avances y opciones que fueran surgiendo.

Las iteraciones que se plantearon desde un inicio de este proyecto están basadas en las que se proponen en algunas de las metodologías ágiles [6], en las que las reuniones del equipo de desarrollo toman una importancia considerable. La frecuencia de las reuniones se estableció al inicio del proyecto en dos semanas, tiempo en el que los desarrolladores implementaban los requisitos estipulados para revisarlos en la siguiente reunión.

En cada una de las iteraciones en primer lugar había una reunión con todo el grupo de desarrollo para evaluar el prototipo de la iteración anterior (si lo había) de manera que el cliente pudiera tener software funcionando al final de cada una de las iteraciones y definir los requisitos que el sistema debía cumplir en la siguiente reunión.

Al finalizar las reuniones, se ha proporcionado a los desarrolladores una lista de tareas a realizar en un tiempo estipulado, de manera que se pueda verificar qué objetivos se han cumplido y cuáles no.

Para poder controlar todas estas tareas se ha hecho uso de la herramienta *Doolphy* [7], mediante de la cual es posible crear tareas y asignárselas al desarrollador que debe encargarse de realizarlas. En ella, los desarrolladores pueden indicar qué porcentaje han realizado de cada tarea e indicar el número de horas utilizado para ello.

1.5 GLOSARIO DE TÉRMINOS Y ABREVIATURAS

En este apartado se recoge un glosario con los términos utilizados en el presente documento, explicando brevemente su significado con el fin de facilitar la lectura del mismo.

1.5.1 GLOSARIO DE TÉRMINOS

Android: Sistema operativo basado en *Linux* propiedad de *Google*. Está orientado a su uso en dispositivos móviles tales como tabletas o teléfonos inteligentes.

App: Forma popularmente aceptada para denominar a aplicación orientadas a los dispositivos móviles, tales como tabletas o teléfonos inteligentes.

AulaGlobal2: Plataforma propiedad de la *Universidad Carlos III de Madrid* que permite a profesores y a alumnos el intercambio de materiales relacionados con la enseñanza.

Doolphy: Aplicación Web que permite la gestión de proyectos software.

Eclipse: Entorno de desarrollo multiplataforma de código abierto que se utiliza para el desarrollo del software.

Facebook: Red social propiedad de Mark Zuckerberg.

Java: Lenguaje de programación creado por Sun Microsystems y actualmente propiedad de Oracle.

Learning Management System o Sistema de Gestión de Aprendizaje: Aprendizaje. Software instalado en un servidor que se encarga de administrar el material ofrecido por el PLE.

Personal Learning Enviroment o Entorno personal de aprendizaje: Es un sistema interactivo que permite a un usuario formarse académicamente de forma autónoma e informal haciendo uso de los recursos compartidos en la plataforma por otros usuarios

Scrum: Metodología de trabajo ágil usada para desarrollos iterativos e incrementales.

Smartphone o Teléfono inteligente: *Teléfono móvil que posee mayor capacidad de cómputo y conectividad que un teléfono móvil convencional, y que habitualmente posee pantalla táctil.*

Tablet o tableta: *Dispositivo móvil similar al Teléfono inteligente pero de tamaño superior. Posee las mismas funcionalidades que el Teléfono exceptuando realizar llamadas.*

Twitter: *Es un servicio basado en el microblogging que permite a sus usuarios escribir pequeños mensajes de 140 caracteres.*

1.5.2 ABREVIATURAS

AVD: *Del inglés Android Virtual Device, Dispositivo Virtual de Android. Es un emulador de Android que puede ser configurado para que posea un hardware (simulado) específico, y que se usa para el desarrollo de aplicaciones junto con el entorno de desarrollo.*

API: *Del inglés Application Programming Interface, Interfaz de programación de aplicaciones. Es una clase abstracta que provee al programador de una funcionalidad básica a implementar, que en muchos casos sirve como intermediaria entre diferentes capas del software.*

DDMS: *Del inglés Dalvik Debug Monitor Server. Es una herramienta de desarrollo que permite la exploración, manipulación y monitorización del dispositivo Android sobre el que se está ejecutando un software.*

LMS: *Del inglés Learning Management System, Sistema de Gestión de Aprendizaje. Software instalado en un servidor que se encarga de administrar el material ofrecido por el PLE.*

PLE: *Del inglés Personal Learning Enviroment, Entorno personal de aprendizaje. Es un sistema interactivo que permite a un usuario formarse académicamente de forma autónoma e informal haciendo uso de los recursos compartidos en la plataforma por otros usuarios*

SDK: *Del inglés Software Development Kit, Kit de desarrollo de software. Conjunto de herramientas de desarrollo que permiten al programador la creación de aplicaciones.*

TFG: Trabajo de Fin de Grado. Adaptación a Bolonia del antiguo proyecto de final de carrera.

XML: Del inglés *eXtensible Markup Language*, lenguaje de marcas extensible. Es un lenguaje de marcas que permite la estructuración de contenido dentro de un documento, facilitando la recuperación y posterior uso del mismo, por el u otro sistema diferente.

1.6 CONTENIDO DE LA MEMORIA

A continuación se hará una breve descripción de las secciones en las que aparecerá dividido el cuerpo del documento:

- **Capítulo 2: Estado del arte:** En este capítulo se describirán los PLE existentes, realizando una breve comparativa de los mismos, se realizará una descripción detallada del sistema operativo *Android*, y por ultimo se explicará la métrica utilizada:
- **Capítulo 3: Requisitos:** En este capítulo contendrá los requisitos de usuario de capacidad y restricción, requisitos funcionales y no funcionales, y matrices de trazabilidad para relacionar requisitos entre sí.
- **Capítulo 4: Diseño:** En este capítulo se presentarán los prototipos de la aplicación móvil, un diagrama de clases y los diagramas de secuencia asociados.
- **Capítulo 5: Pruebas:** En este capítulo se exponen los resultados obtenidos de las pruebas unitarias, de integración y de sistema.
- **Capítulo 6: Conclusiones:** En este capítulo se presentan las conclusiones obtenidas durante la elaboración del trabajo.

CAPÍTULO 2: ESTADO DEL ARTE

2.1 ANÁLISIS DE APLICACIONES WEB Y MÓVIL RELACIONADAS CON LOS OBJETIVOS ESTABLECIDOS	10
2.1.1 WC3SCHOOLS	10
2.1.2 BUSUU	11
2.1.3 BLACKBOARD	12
2.1.4 KHAN ACADEMY	13
2.1.5 WIKILEARNING	14
2.1.6 SLIDESHARE	15
2.2 ENTORNO DE DESARROLLO MÓVIL SELECCIONADO	17
2.2.1 ENTORNOS MÓVILES.....	17
2.2.2 VISIÓN GENERAL SOBRE ANDROID	18
2.2.3 ESTRUCTURA DE ANDROID	20
2.2.4 ENTORNO DE DESARROLLO INTEGRADO.....	21
2.3 DESARROLLO ÁGIL COMO MÉTODO PARA REALIZAR EL PROYECTO.....	22
2.3.1 THE AGILE ALLIANCE	22
2.3.2 SCRUM.....	22
2.4 CONCLUSIONES DEL ANÁLISIS	24

Como primer paso antes de afrontar un proyecto de estas características es necesario realizar un estudio de mercado en base al contexto sobre el que se operará, y a las aplicaciones existentes, ya sean de carácter móvil o Web, que existen en el mismo, analizando sus características para su posterior comparación con las nuestras, con la intención de mejorar lo ya existente.

En este capítulo se expone la información obtenida del estudio del mercado tomando como objetivo del mismo las aplicaciones existentes con fines similares a la nuestra, describiendo sus características fundamentales y sus puntos más favorables y desfavorables comparados con nuestra aplicación. A continuación se explicará la plataforma sobre la que se ha realizado el trabajo, haciendo una breve introducción a sus características fundamentales y a continuación a las características propias de la versión utilizada. Para concluir el capítulo se explicará la forma en la que se ha utilizado la metodología de desarrollo ágil usada para la elaboración de este trabajo.

El estudio del mercado se ha realizado obteniendo información a través del buscador Web *Google*, para aplicaciones Web, y utilizando el buscador de *Google Play* para aplicaciones móvil, para su posterior análisis y síntesis. Los términos usados para las búsquedas procesadas se ofrecen a continuación.

Todas las búsquedas realizadas han estado orientadas a la localización de aplicaciones tanto Web como de carácter móvil para terminales *Android* que permitan el aprendizaje en línea del usuario.

En buscador Web **Google** se han realizado las siguientes búsquedas:

- *aprendizaje en línea*
- *e-learning*
- *learn*

Sobre la tienda de *Android Google Play* se han realizado las siguientes búsquedas:

- *aprendizaje*
- *e-learning*
- *learn*

Todas las búsquedas se realizaron entre los días 15 y 20 de Diciembre del año 2012, y se revisaron el día 12 de Enero de 2013, y por lo tanto las aplicaciones han sido analizadas en base a su estado en esta última fecha.

2.1 ANÁLISIS DE APLICACIONES WEB Y MÓVIL RELACIONADAS CON LOS OBJETIVOS ESTABLECIDOS

En primer lugar se deben dejar claras las principales características que nuestra aplicación posee, y que buscaremos en aplicaciones similares, para su posterior comparación y valoración objetiva atendiendo a sus puntos más fuertes y débiles. Los rasgos buscados en otras aplicaciones son:

- La aplicación permite compartir material docente u académico.
- La aplicación permite el control de alumnos que siguen una asignatura o curso, y de asignaturas o cursos que sigue un alumno.
- El acceso a la aplicación está limitado solo a usuarios registrados.

A continuación se detallan las aplicaciones más relevantes que se han encontrado y sus características más importantes en relación a este trabajo:

2.1.1 WC3SCHOOLS

Dentro de las búsquedas realizadas en *Google Play* fue de los primeros resultados en aparecer para aplicaciones de aprendizaje en línea.

Esta aplicación no es más que la Web de *wc3schools.com* [8] adaptada a una aplicación móvil, y permitiendo acceder a los contenidos en estado offline del terminal. Al igual que su homónima en Web esta aplicación permite al usuario formarse en temas orientados al desarrollo de aplicaciones.

En rasgos generales la aplicación ofrece al usuario una pantalla de inicio, estructurada igual que una Web, en la que se le permite seleccionar la materia de la que quiere aprender, y se le redirecciona al material seleccionado. Los temas de estudio se encuentran distribuidos en una barra de navegación lateral, en el margen izquierdo de la aplicación.

La fluidez de la misma es muy buena, dado que todo el contenido que maneja se encuentra en local del dispositivo móvil, lo cual agiliza mucho la carga del mismo.

La apariencia de la aplicación, aunque atractiva, está mal optimizada para terminales móviles, al menos para teléfonos con pantalla inferior a 4', ocultando parte del contenido permanentemente, y obligando al usuario a realizar scrolls laterales para visualizarlo.

Todos los contenidos ofrecidos por esta aplicación se encuentran embebidos en la instalación de la misma, siendo solo necesaria conexión a Internet para su descarga desde *Google Play*.



Figura 2.1.1: Wc3schools

2.1.2 BUSUU

La aplicación *Busuu* [9] permite al usuario el aprendizaje en línea de idiomas. *Busuu* tiene múltiples aplicaciones, en más de 12 idiomas, y el estudio concretamente se ha realizado sobre permite al usuario el aprendizaje del alemán.

La aplicación ofrece al usuario una pantalla de inicio en la que se le muestran opciones para identificarse, acceder a un curso, continuar un curso ya empezado, o revisar errores cometidos en los cursos realizados. Todas las opciones pueden ser usadas con o sin identificarse en la aplicación, siendo esta opción útil solo para almacenar el contenido de los cursos completados, y tener cierto control sobre los mismos, principalmente de cara al usuario final.

Todos los contenidos ofrecidos por esta aplicación se encuentran embebidos en la instalación de la misma, siendo solo necesaria conexión a Internet para su descarga desde *Google Play*.



Figura 2.1.2: Busuu

2.1.3 BLACKBOARD

BlackBoard [10] es una aplicación con vínculos docentes que permite al usuario acceder a los cursos ofrecidos por su centro de estudios, y acceder a determinados contenidos que este ofrece de forma virtual. La aplicación de *BlackBoard* es un cliente que se comunica con un sistema de gestión de contenidos para el aprendizaje del cual extrae la información compartida por parte del centro de enseñanza.

BlackBoard ofrece al usuario un formulario de búsqueda, que permite localizar el centro de estudios del usuario, con el fin de permitirle acceder a los materiales que su centro ofrezca mediante este sistema. No obstante hay que señalar que el uso de *BlackBoard* no está muy extendido, y que para realizar las pruebas referentes a este trabajo ha sido necesario usar una demo que el propio programa facilita en caso de no permitir el acceso al centro solicitado.

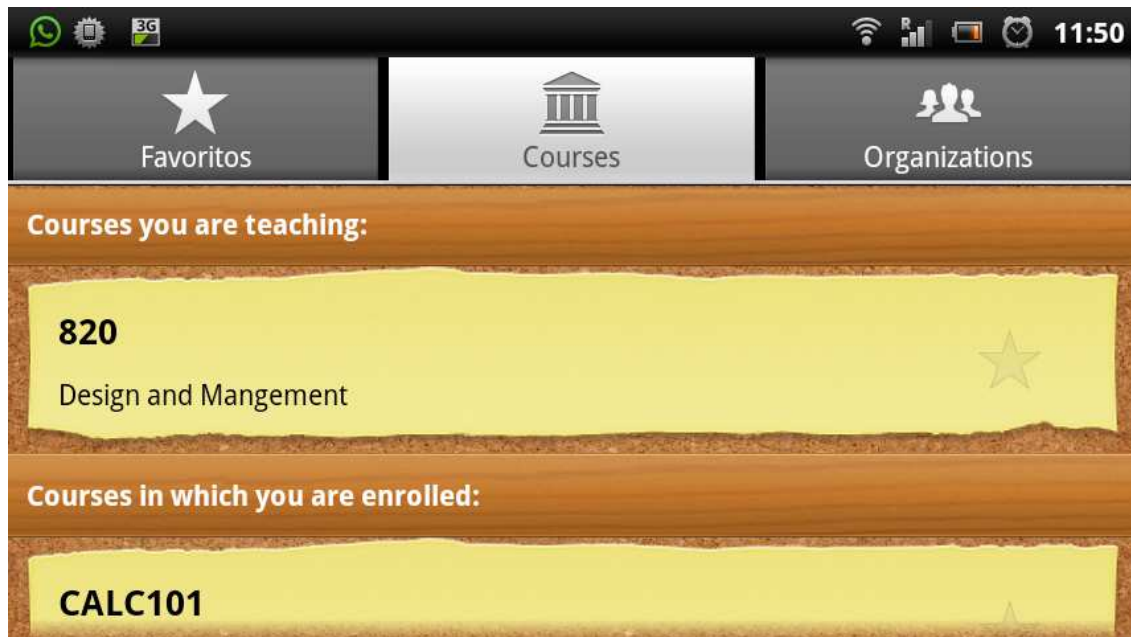


Figura 2.1.3: Blackboard

2.1.4 KHAN ACADEMY

Khan academy [11] es una aplicación que permite al usuario formarse en diversos temas, agrupados en 5 grandes ramas: “*Math*”, “*Science*”, “*Humanities & Other*”, “*Test Prep*”, “*Talks and Interviews*”, cada una de las cuales posee decenas de subtemas más específicos sobre los que se puede conseguir formación.

Khan academy ofrece al usuario la posibilidad de formarse mediante video tutoriales distribuidos por dos vías, ya sea por *Youtube* o directamente desde la aplicación, previa descarga del video.

Tanto la aplicación como el material ofrecido en la misma se encuentran íntegramente en inglés.

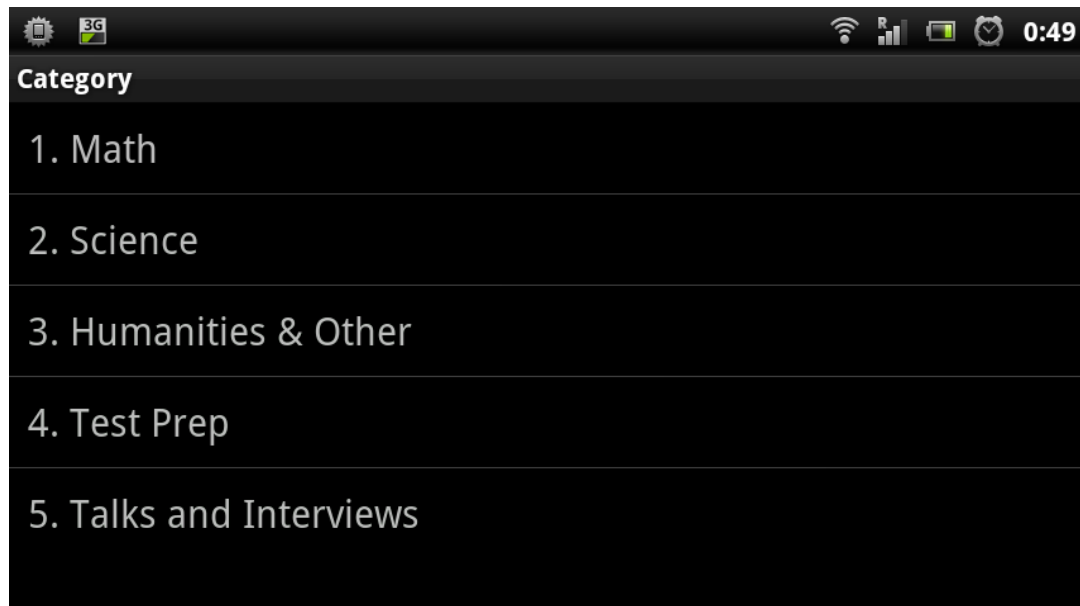


Figura 2.1.4: Khan academy

2.1.5 WIKILEARNING

Wikilearning [12] es una plataforma de aprendizaje en línea en forma de página Web. De todo el conjunto de sitios Web con este tipo de fines *Wikilearning* es de los más reconocidos de habla hispana.

Wikilearning ofrece al usuario contenidos para el estudio en forma de *wiki*, desplegando el contenido de cada *wiki* de una manera muy sencilla, como si de un *tópico* de un foro se tratase.

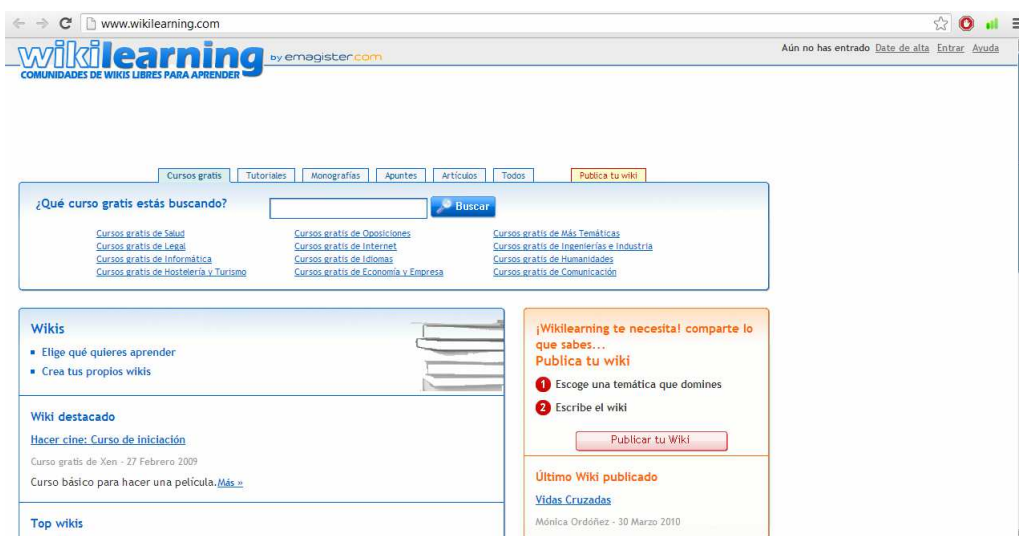


Figura 2.1.5: Wikilearning

2.1.6 SLIDESHARE

Slideshare [13] es una plataforma Web de aprendizaje que ofrece material de aprendizaje en forma de presentaciones, similares a un *PowerPoint*. Dichos contenidos suelen ser compartidos por otros usuarios, que lo suben a la Web como aporte para el crecimiento de la comunidad, y para que otros usuarios lo usen.

Uno de los detalles más atractivos que posee *Slideshare* y que pocos sitios Web de estas características posee es la conexión con redes sociales para compartir información o interconectarse con amigos. *Slideshare* permite conectarse con *Facebook* para buscar amigos, y compartir cualquier contenido que ofrece la página a través de *Twitter*, *Linkedin*, *Google+* y *Facebook*.



Figura 2.1.6: Slideshare

Tras explicar las características de cada una de las aplicaciones mencionadas se va procederá a realizar un análisis comparativo entre ellas atendiendo a las siguientes características:

- **Permite el acceso al sistema mediante la identificación del usuario:** La aplicación permite el acceso a la misma mediante la identificación del usuario frente al sistema.
- **Permite el acceso al sistema utilizando la identidad social del usuario:** La aplicación permite el acceso a la misma utilizando como método de identificación frente al sistema la identidad social que el usuario posee en alguna red social.
- **Permite la suscripción a cursos:** La aplicación permite al usuario la suscripción a cursos que este crea convenientes, permitiendo el acceso a los contenidos de los cursos suscritos de forma sencilla.

- **Está disponible para terminales Android:** La aplicación está disponible para su uso en terminales Android (teléfonos o tabletas) siendo posible su adquisición mediante *Google Play*.
- **Está optimizada para terminales Android:** La aplicación está optimizada para su uso en terminales Android, poseyendo unas características de resolución y usabilidad aceptables para un uso cómodo por parte del usuario.
- **Requiere conexión a internet para el acceso a contenidos:** La aplicación requiere de conexión a Internet permanente para el acceso a los contenidos de estudio ofrecidos.
- **Realizado como complemento a clases presenciales:** La aplicación tiene un respaldo docente en forma de clases presenciales en algún centro o Universidad.

El resultado de dicha comparación se puede consultar en la siguiente tabla:

	Permite identificación del usuario	Permite identificación por identidad social	Suscripción a cursos	Disponible en Android	Optimizada para Android	Requiere Internet	Complemento a clases presenciales
Wc3schools	X	X	X	V	X	X	X
Musuu	V	X	X	V	V	X	X
Wikilearning	V	X	X	X	X	V	X
Slideshare	V	V	V	X	X	V	X
BlackBoard	X	X	X	V	V	V	V
Khan Academy	X	X	X	V	V	V	X

Tabla 2.1: Comparativa aplicaciones

2.2 ENTORNO DE DESARROLLO MÓVIL SELECCIONADO

En esta sección se analizará el estado actual los diferentes entornos móviles existentes, se hará una breve introducción a *Android* y a su organización interna, y por último se describirá el entorno móvil seleccionado, y las razones por las cuales se ha escogido este y no otro de los existentes.

2.2.1 ENTORNOS MÓVILES

Los entornos móviles más comúnmente usados en el primer trimestre del año 2012 fueron [14], por orden de importancia, *Symbian*, *iOS*, *Android*, *BlackBerry*, y otros.

Tal y como se muestra en la siguiente figura el número de terminales móviles con sistema operativo *Android* ronda el 25%, ocupando una tercera plaza, por detrás de *Symbian* y *iOS*.

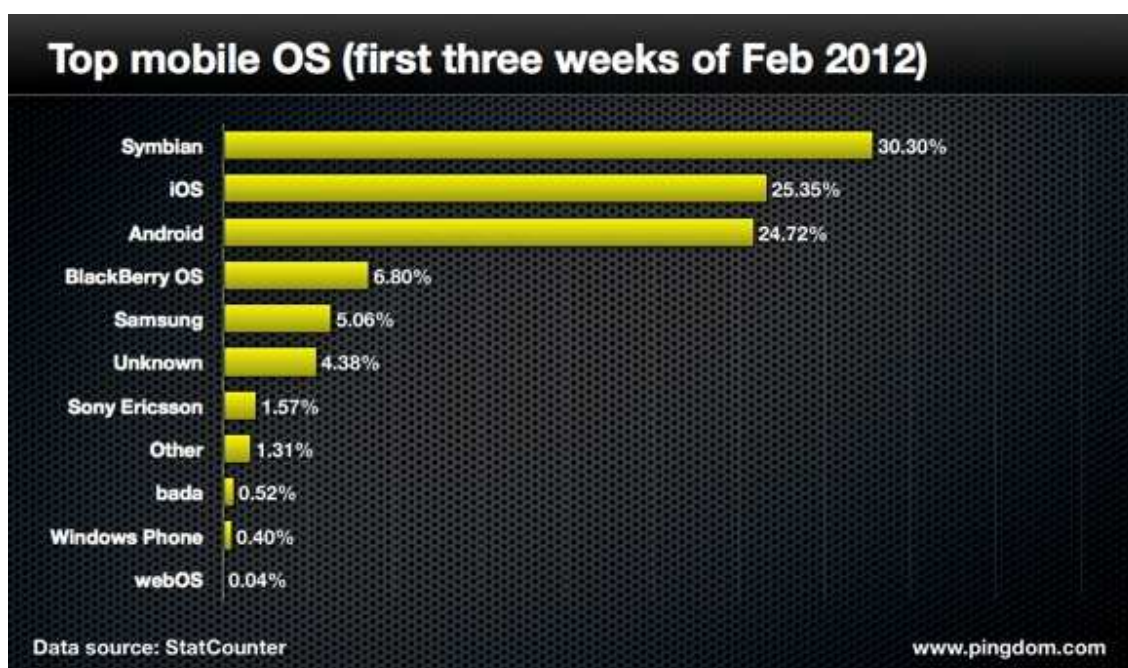


Figura 2.2.1.1: Terminales operativos 2012

La explicación, pese a que resulte sorprendente, a que *Symbian* posea más terminales activos en el mundo que *Android* radica en que la gran mayoría de los terminales móviles antiguos poseen sistemas operativos basados en *Symbian*. Estos terminales suelen ser más duros que los actuales, y por lo tanto siguen formando parte del volumen de terminales activos en el mundo, abultando la estadística a favor de *Symbian*.

No obstante, atendiendo a los datos de consumo del año 2012, en España de todo el volumen de teléfonos inteligentes comprados un 59% poseía sistema operativo *Android*. Dichos datos se reflejan en el siguiente gráfico.

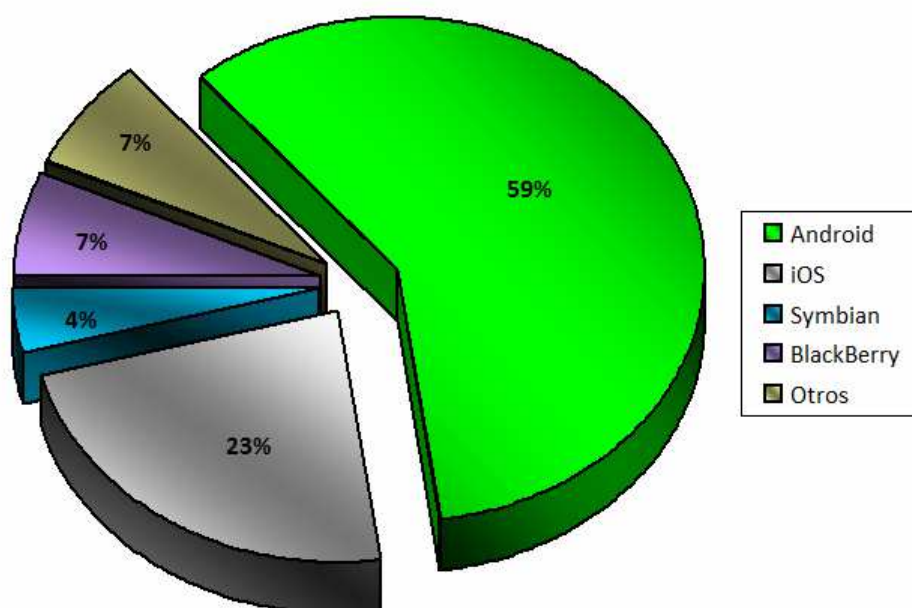


Figura 2.2.1.2: Ventas terminales 2012

Atendiendo a los datos de consumo del último año se ha decidido elegir *Android* como plataforma para albergar la aplicación móvil objeto de este trabajo.

2.2.2 VISIÓN GENERAL SOBRE ANDROID

El entorno de desarrollo móvil escogido para este trabajo ha sido el sistema operativo *Android* dado que, como se explicó anteriormente, es el entorno con más altas en el año 2012, y la intención de este trabajo es llegar al mayor número usuarios posible.

Las versiones del sistema operativo *Android* se nombran con un número de versión y un nombre de la misma, y en la actualidad abarcan desde la versión 1.0 a la actual versión 4.2. En la siguiente tabla se recogen todas las versiones de *Android* y su nombre característico.

Nombre	Versión
“Apple Pie”	Android 1.0
“Banana Bread”	Android 1.1
“Cupcake”	Android 1.5
“Donut”	Android 1.6
“Éclair”	Android 2.0-2.1
“Froyo”	Android 2.2

“Gingerbread”	Android 2.3
“Honeycomb”	Android 3.0
“Ice Cream Sándwich”	Android 4.0
“Jelly Bean”	Android 4.1-4.X

Tabla 2.2.2: Versiones Android

El sistema operativo *Android* fue lanzado al mercado el día 21 de Octubre de 2008 con su versión 1.0. *Android* es un sistema operativo basado en *Linux* ideado para ser utilizado en teléfonos inteligentes y tabletas, y desarrollado por *Open Handset Alliance*, la cual es liderada por *Google*.

La versión más novedosa que posee *Android*, y que actualmente se está comercializando, es la versión 4.2 con nombre en clave “*Jelly Bean*”, la cual es una versión mejorada de la versión 4.0 con nombre en clave “*Ice Cream Sándwich*”, versión que ya introducía mecanismos de seguridad robustos para la protección del terminal, y más concretamente la información que este contenía. La fragmentación que posee *Android* es muy elevada, dada la diversidad de dispositivos existentes con limitaciones hardware completamente distintas, y la cantidad de versiones existentes lanzadas al mercado. Esto provoca que de todo el conjunto de terminales *Android* haya grupos delimitados por la versión del terminal. Dicha información se muestra en la siguiente tabla, así como el nivel de API de cada una de ellas, es decir, la funcionalidad que una aplicación puede desarrollar mediante su interfaz de programación interactuando con el sistema.

Plataforma ◆	Nivel de API ◆	% ◆
4.x.x <i>Jelly Bean</i>	16-17	6,7%
4.0.x <i>Ice Cream Sandwich</i>	14-15	27,5%
3.x.x <i>Honeycomb</i>	12-13	1,6%
2.3.x <i>Gingerbread</i>	9-10	50,8%
2.2 <i>Froyo</i>	8	10,3%
2.1 <i>Eclair</i>	7	2,7%
1.6 <i>Donut</i>	4	0,3%
1.5 <i>Cupcake</i>	3	0,1%

Figura 2.2.2: Versiones Android

Desde la versión 3.0 la intención de *Google* era unificar las dos vías de desarrollo de *Android*, que eran tabletas y teléfonos, en una vía común. En la versión 4.0 este proceso de unificación se completó correctamente, obteniendo una versión global para todos los dispositivos *Android* del mercado.

Para el desarrollo de este trabajo se ha escogido la versión 4.0.3 por dos razones, la primera gran cuota de uso, dentro de las versiones existentes de *Android*. La segunda es por su versatilidad, la cual le permite funcionar en tabletas o teléfonos.

2.2.3 ESTRUCTURA DE ANDROID

El sistema operativo *Android* posee una serie de componentes [15] que se muestran en la siguiente figura, y que se explican a continuación.



Figura 2.2.3: Estructura Android

Aplicaciones

Conjunto de aplicaciones que vienen preinstaladas en el sistema operativo *Android* y que facilitan al usuario determinadas acciones como son: consultar correo electrónico, acceder a la galería de fotos, añadir y consultar contactos, etc. Toda esta funcionalidad básica está programada en Java.

Framework de aplicaciones

Conjunto de *APIs* que pueden ser utilizadas por los desarrolladores y que facilitan el uso de funcionalidades más complejas, y que habitualmente se llevan a cabo en el común de las aplicaciones. Dichas *APIs* facilitan al desarrollador el uso de elementos como son el teléfono, las actividades en ejecución, servicios, notificaciones etc.

Bibliotecas

Android posee una serie de bibliotecas *C/C++* que son utilizadas por algunos componentes del sistema operativo, y que están expuestas al uso de los desarrolladores.

Android en tiempo de ejecución

Cada aplicación *Android* se ejecuta en una región de memoria independiente y fijada por el sistema operativo, realizando su propia instancia de la máquina virtual *Dalvik*.

El *kernel* de *Linux* puede manejar múltiples instancias de la máquina virtual *Dalvik*, una por cada aplicación, manejando cada una de estas como un hilo de ejecución independiente. A este proceso de ejecución basado en el uso de una pequeña región de memoria independiente, fijada por el sistema operativo, y en nueva instancia de la máquina virtual se le conoce como *sandboxing*, y provee al sistema operativo de mayor seguridad y versatilidad.

Kernel de Linux

Android depende de *Linux* para servicios del *kernel* del sistema, tales como la gestión de la memoria y procesos, la pila de red, el modelo de controlador o la seguridad. El *kernel* también actúa como intermediario entre el hardware y las aplicaciones, aportando abstracción en la comunicación entre ambos.

2.2.4 ENTORNO DE DESARROLLO INTEGRADO

El entorno de desarrollo usado para la elaboración de este trabajo ha sido *Eclipse Indigo*, ya que ha sido el entorno de desarrollo más comúnmente utilizado a lo largo de la carrera, factor que le convirtió en la opción más cómoda, y además ya lo tenía instalado en mi sobremesa preparado para funcionar.

Para proceder al desarrollo del trabajo ha sido necesario instalar el *SDK* de *Android*, configurar su *ADT* y configurar un *AVD*, sobre el que se han realizado todas las pruebas de ejecución de la aplicación desarrollada.

Para finalizar cabe destacar dos herramientas que el entorno de desarrollo nos ofrece, y que han sido de gran ayuda en la depuración de errores, como son el *DDMS* y el *LogCat*. *DDMS* nos permite la exploración del *AVD* sobre el que se realizan las pruebas de ejecución de nuestra aplicación, pudiendo ver en tiempo real como manipula, modifica y crea ficheros. Por otro lado *LogCat* nos muestra trazas de ejecución de la aplicación,

diferenciadas por colores, tipos y paquetes, lo cual ha sido también de gran ayuda para localizar errores y excepciones.

2.3 DESARROLLO ÁGIL COMO MÉTODO PARA REALIZAR EL PROYECTO

Las metodologías ágiles son métodos para el desarrollo de software basados en un desarrollo iterativo e incremental, donde todo el desarrollo es llevado a cabo por grupos de trabajadores autoorganizados y multidisciplinarios. Comparando este tipo de desarrollos con los convencionales podemos apreciar que un desarrollo ágil provee a un proyecto de mayor flexibilidad a cambios, y de una ideología de colaboración entre sus trabajadores.

Otro de los aspectos clave de las metodologías ágiles es que prevalece el trato de las personas cara a cara, hablando entre ellas para afrontar partes del desarrollo, frente a los documentos estáticos existentes en las metodologías convencionales de desarrollo software. Esta mentalidad a la hora de afrontar el desarrollo facilita la comunicación entre los trabajadores, lo cual concluye en un desarrollo de calidad con un menor coste.

2.3.1 THE AGILE ALLIANCE

The Agile Alliance es una organización fundada en el año 2001, sin ánimo de lucro, creada y dedicada para promover el desarrollo ágil de software, frente al desarrollo convencional del mismo. *The Agile Alliance* desarrolló un manifiesto en el que se establecen los puntos básicos de su filosofía de desarrollo. A continuación se ofrece un breve resumen de las ideas más importantes del mismo:

- La gente es el principal factor de éxito de un proyecto software. Es preferible crear un buen equipo y que este configure su entorno de desarrollo en base a sus gustos o necesidades, a crear un buen entorno y esperar que el equipo se adapte automáticamente.
- Es preferible desarrollar software que funciona frente a una documentación de calidad. Solo se deben crear documentos cuando realmente hacen falta, y han de ser claros y concisos.
- El cliente debe formar parte del desarrollo. Si este mantiene comunicación con el equipo de desarrollo regularmente el proyecto tiene asegurado el éxito.
- La planificación del proyecto ha de ser flexible y abierta, facilitando posibles cambios o mejoras a lo largo del desarrollo del mismo.

2.3.2 SCRUM

Para concluir se va a hacer referencia a la metodología de desarrollo ágil utilizada en este trabajo, *Scrum* [16].

Uno de los principales factores que hacen que *Scrum* sea una metodología atractiva para el desarrollo de software es su simplicidad, la cual requiere de muy poco trabajo

comprenderla, y la segunda es su capacidad para maximizar el desarrollo de software en periodos cortos de tiempo.

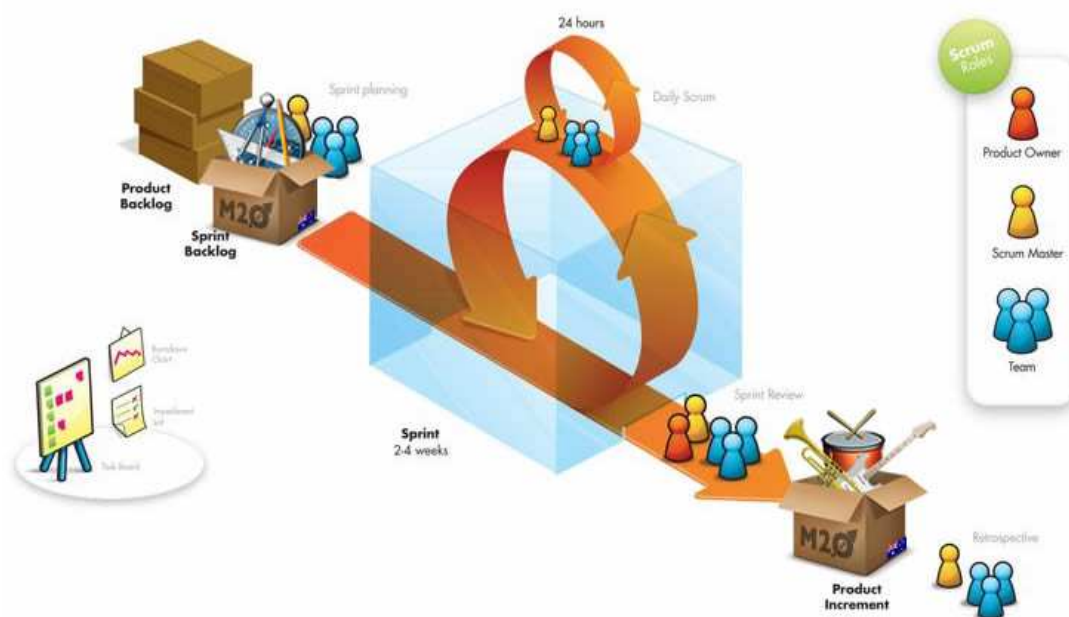


Figura 2.3.2: Scrum

Los beneficios [16] de utilizar *Scrum* como método de desarrollo ágil son los siguientes:

- **Cumplimiento de expectativas:** El cliente establece sus expectativas indicando el valor que le aporta cada requisito/historia del proyecto, el equipo los estima y con esta información establece su prioridad. De manera regular, en las demos de *Sprint* se comprueba que efectivamente los requisitos se han cumplido y se transmite feedback (retroalimentación) al equipo.
- **Flexibilidad a cambios:** Alta capacidad de reacción ante los cambios de requerimientos generados por necesidades del cliente o evoluciones del mercado.
- **Reducción del Time to Market:** El cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que esté finalizado por completo.
- **Mayor calidad del software:** La metódica de trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.
- **Mayor productividad:** Se consigue entre otras razones, gracias a la eliminación de la burocracia y a la motivación del equipo que proporciona el hecho de que sean autónomos para organizarse.
- **Maximiza el ROI:** Producción de software únicamente con las prestaciones que aportan mayor valor de negocio gracias a la priorización por retorno de inversión.

- **Predicciones de tiempos:** Mediante esta metodología se conoce la velocidad media del equipo por *sprint* (los llamados puntos historia), con lo que consecuentemente, es posible estimar fácilmente para cuando se dispondrá de una determinada funcionalidad que todavía está como trabajo pendiente.
- **Reducción de riesgos:** El hecho de llevar a cabo las funcionalidades de más valor en primer lugar y de conocer la velocidad con que el equipo avanza en el proyecto, permite despejar riesgos eficazmente de manera anticipada.

2.4 CONCLUSIONES DEL ANÁLISIS

En función de los resultados que hemos obtenido del análisis de otras aplicaciones nos hemos percatado de varios hechos que sería conveniente destacar. El primero es que, como tal, no existe ninguna aplicación móvil que tenga un propósito similar, es decir, facilitar al alumno el acceso al material docente de sus estudios, y por otro lado, facilitar al profesor la compartición del material docente de su asignatura entre los alumnos que componen su clase. Todas las aplicaciones móviles existentes ofrecen una enseñanza en línea independiente y en la mayoría de los casos no vinculada a ninguna titulación, centro u organización. En el caso concreto de *BlackBoard* se ofrece un vínculo con el centro de estudios, pero no se aprecia una relación virtual entre profesor y alumnos, siendo el acceso a los contenidos docentes algo anónimo y sin control. Además una mayoría de las aplicaciones analizadas ofrecen contenidos estáticos, existentes en el instalador de la aplicación, en contraposición al objeto de este trabajo, que ofrece contenidos a los alumnos cambiantes y coordinados por la figura de un profesor, mediante un LMS.

En el desarrollo de esta aplicación móvil queremos utilizar una metodología de desarrollo ágil, puesto que en un principio, los requisitos de la aplicación no están definidos. En este caso, se elige Scrum que es la más adecuada a este tipo de desarrollos software, pero se ha adaptado para que las iteraciones sean de dos semanas. En esas iteraciones se parte de una reunión de los miembros del equipo entorno a la última versión del producto (si la hay) para definir las tareas a realizar durante la iteración por cada uno de los miembros del equipo de manera que al final exista una nueva versión del producto.

Durante el tiempo en que se llevan a cabo esas tareas, se hacen reuniones diarias para comprobar el avance del trabajo, para comentar cuestiones técnicas que no hubieran quedado claras en la reunión anterior o para plantear alternativas que se planteen en la siguiente reunión común.

CAPÍTULO 3: ESPECIFICACIÓN DE REQUISITOS

3.1 INTRODUCCIÓN	27
3.2 DESCRIPCIÓN GENERAL DEL SISTEMA A DESARROLLAR	27
3.2.1 PERSPECTIVAS DEL PRODUCTO	27
3.2.2 FUNCIONES DEL PRODUCTO	28
3.2.3 CARACTERÍSTICAS DE LOS USUARIOS.....	29
3.2.4 INTERACCIÓN CON OTROS SISTEMAS	29
3.2.5 RESTRICCIONES	29
3.2.6 SUPOSICIONES Y DEPENDENCIAS.....	30
3.3 REQUISITOS ESPECÍFICOS	30
3.3.1 REQUISITOS FUNCIONALES	32
3.3.2. INTERFACES EXTERNAS	36
3.3.3 REQUISITOS DE RENDIMIENTO.....	42
3.3.4 RESTRICCIONES DE DISEÑO	42
3.3.5 ATRIBUTOS DEL SISTEMA.....	44

3.1 INTRODUCCIÓN

En este capítulo se realiza la Especificación de Requisitos Software (*ERS*). En ella se muestran los requisitos necesarios para el correcto funcionamiento de la aplicación. Esta *ERS* esta dirigida a cualquier persona que desee saber acerca de la cumplimentación de requisitos de esta aplicación.

Para el desarrollo de la especificación de requisitos se ha utilizado *Revisión de Métrica v3* [17] como documento de referencia.

El contexto en el que se desarrolla la aplicación es el momento de expansión tecnológica que permite poseer a casi todos los ciudadanos un teléfono inteligente o una tableta, y se decide crear una aplicación que sirva como apoyo a la educación, permitiendo la compartición de material docente en terminales *Android* en primera instancia, e *iOS* próximamente.

La estructura que poseerá este capítulo, atendiendo a los puntos que se van a tratar en el mismo, es la siguiente:

- 3.2 Descripción general del sistema a desarrollar: En esta sección se realizará una descripción del sistema a desarrollar, atendiendo a sus expectativas y funciones, además de exponer los usuarios que potencialmente utilizarán la aplicación, las interacciones realizadas entre la aplicación y otros sistemas, y para concluir las restricciones y supuestos que la aplicación posee.
- 3.3 Requisitos: En esta sección se realizará la especificación de requisitos software, la cual está compuesta por requisitos funcionales, interfaces externas, rendimiento y restricciones.

3.2 DESCRIPCIÓN GENERAL DEL SISTEMA A DESARROLLAR

En esta sección se describen todos los factores que pudieran alterar el estado del producto final a desarrollar, y por lo tanto de sus requisitos. En primer lugar se describirá el contexto de la aplicación, atendiendo al público que la utilizará y en que condiciones, para posteriormente en el punto 3.3 proceder a la definición formal de requisitos.

3.2.1 PERSPECTIVAS DEL PRODUCTO

Como ya se comentó anteriormente, la aplicación *MoLe* es un módulo de un *PLE* diseñado para la compartición de material docente entre profesores y alumnos, de forma complementaria a otras plataformas de aprendizaje, y que opera de forma independiente a la plataforma, posibilitando su uso mediante teléfonos inteligentes o tabletas, permitiendo la posibilidad de acceder a dichos contenidos mientras se asiste a una clase presencial, todo ello en tiempo real.

La aplicación permite al usuario suscribirse a las asignaturas que desee, permitiéndole el acceso a los contenidos que estas ofrezcan mediante su tablón. De modo

opuesto y complementario, los profesores puede agregar o modificar contenidos al tablón de su asignatura, y consultar la lista de alumnos que se encuentran suscritos a la misma, permitiendo excluir a cualquiera de la misma, siempre y cuando fuera necesario.

El acceso a la aplicación se realiza usando los datos de usuario de cuentas de *Twitter* activas.

3.2.2 FUNCIONES DEL PRODUCTO

Las principales funciones de la aplicación *MoLe* son las siguientes:

Como alumno

- Identificarse en la aplicación.
- Acceder al listado de cursos a los que se está suscrito.
- Suscribirse a un nuevo curso.
- Borrar suscripciones a cursos.
- Acceder al tablón de contenidos de un curso suscrito.
- Descargar materiales compartidos en el tablón de un curso suscrito.

Como profesor

- Identificarse en el sistema.
- Mostrar la lista de alumnos suscritos a la asignatura.
- Borrar alumnos suscritos a la asignatura.
- Visualizar el contenido del tablón de la asignatura.
- Modificar el contenido del tablón de la asignatura.
- Borrar el contenido del tablón de la asignatura.
- Añadir nuevas entradas al tablón de la asignatura.
- Compartir material académico mediante el tablón de la asignatura.
- Compartir recursos localizados en el terminal móvil o en un recurso remoto.

3.2.3 CARACTERÍSTICAS DE LOS USUARIOS

Los usuarios que utilizarán la aplicación poseerán dos características fundamentales. La primera es que poseerán un teléfono inteligente o una tableta con sistema operativo *Android*. La segunda es que están relacionados con la educación, como profesores o alumnos, siendo el marco más común de uso para la aplicación la Universidad, ya que para el funcionamiento del sistema se requiere de profesores que creen asignaturas y compartan contenidos, y de alumnos que los utilicen. Faltando cualquiera de los dos elementos el sistema estaría incompleto y no tendría sentido. Por esta razón la aplicación está estrechadamente ligada con la impartición de estudios oficiales, y generalmente con soporte presencial. No obstante existen más modelos educativos en los que la aplicación puede ser usada correctamente, pero es necesaria, como se detalló anteriormente, la existencia de las dos figuras, profesor y alumno.

3.2.4 INTERACCIÓN CON OTROS SISTEMAS

La aplicación va a interactuar con otros sistemas ajenos e independientes al desarrollo acometido en este trabajo. Por esta razón el buen funcionamiento de dichos sistemas queda fuera del marco de las obligaciones relacionadas con el desarrollo de esta aplicación. Los sistemas con los que la aplicación se comunica, y la función que estos llevan a cabo en el funcionamiento de este trabajo es la siguiente:

- **Twitter:** Este sistema es el pilar básico para el funcionamiento de esta aplicación, dado que mediante este sistema se realiza el acceso a la aplicación, se listan las suscripciones, y se realizan todas las altas y bajas de suscripciones, en todas las modalidades ofrecidas, tanto para profesores como para alumnos.
- **Servidor SEL:** Este sistema hará las funciones de *LMS* de nuestro *PLE*, y se encargará de almacenar los ficheros a compartir, así como la estructura e información de los tabloneros para cada asignatura.

3.2.5 RESTRICCIONES

Tanto la interfaz como la funcionalidad desarrollada por la aplicación han sido desarrollados bajo el *API-15* del *SDK* de *Android*, referente a la versión 4.0.3 “*Ice Cream Sandwich*”, por lo que para el correcto funcionamiento de la aplicación es necesario ejecutarla sobre un dispositivo con una versión igual o superior a la utilizada para el desarrollo.

La aplicación *MoLe* posee una *API* para la comunicación con el *LMS*, la cual ha sido implementada en base al sistema con el que se realiza la comunicación actualmente. No obstante si se establece otro sistema como *LMS* se deberá implementar esta acorde al sistema con el que se desee la interconexión. Para el correcto funcionamiento de la aplicación se deberá respetar el formato de la *API* en todo momento.

El lenguaje de programación usado para todo el desarrollo de la aplicación ha sido *Java*, siendo el lenguaje indicado para el desarrollo de aplicaciones *Android*.

Para el correcto funcionamiento de la aplicación el terminal sobre el que se ejecute deberá poseer espacio de almacenamiento libre.

La velocidad de ejecución de la aplicación es dependiente del terminal sobre el que se ejecute y las condiciones de tráfico de red existentes, así como el correcto funcionamiento de la plataforma *Twitter* y el *LMS*. Cualquier problema de red o de tráfico provocará funcionamientos lentos de la aplicación.

La seguridad de la aplicación es dependiente de la seguridad de *Twitter*, siendo la plataforma que brinda el acceso a *MoLe*, por lo que no se han tomado medidas de seguridad extra, confiando en la buena praxis de *Twitter* en este aspecto, y de la seguridad del servidor de *SEL*, confiando en la capa de seguridad existente en el mismo.

La aplicación no almacenará datos de acceso, ni contraseñas de *Twitter* del usuario que realiza el acceso a *MoLe*.

El acceso a *Twitter* se debe realizar en cada uso de la aplicación *MoLe* por la caducidad del objeto de sesión que *Twitter* genera y que permite el uso de sus recursos en *MoLe*.

La aplicación necesita en todo momento conexión a Internet para su funcionamiento.

3.2.6 SUPOSICIONES Y DEPENDENCIAS

La aplicación *MoLe* hace uso de los recursos existentes en el servidor de *SEL*, por lo que si el servicio dejara de estar operativo la aplicación no permitirá acceder al contenido de los tableros, ni manipular el contenido existente en ellos, dado que no podría acceder al mismo.

3.3 REQUISITOS ESPECÍFICOS

En esta sección se ofrece la especificación de requisitos desarrollada para la aplicación *MoLE*, mediante la cual se detallan las características que la aplicación debe poseer con la intención de satisfacer las necesidades impuestas a este desarrollo. Mediante esta especificación de requisitos se permite a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas comprobar mediante pruebas si el sistema satisface o no los requisitos impuestos en esta especificación.

En la sección 3.3.1 se muestran los requisitos funcionales especificados para la aplicación *MoLE*, siendo el objetivo de estos describir la funcionalidad que la aplicación es capaz de desarrollar.

En la sección 3.3.2 se muestran los requisitos asociados a las interfaces que posee la aplicación, siendo estas todos aquellos lugares que permiten al usuario una interacción con

la misma, ya sea directa o indirectamente, es decir, sobre la que el usuario realizar acciones, o que solo le muestran información. Además se mostrarán los requisitos asociados a las interfaces externas, producto de aplicaciones ajenas a la nuestra que son utilizadas en el uso de la misma, y que contribuyen al funcionamiento de la aplicación.

En la sección 3.3.3 se mostrarán los requisitos que afectan al rendimiento de la aplicación.

En la sección 3.3.4 se mostrarán las restricciones de diseño existentes en la aplicación, y que limitan ciertos aspectos del funcionamiento de la misma.

Para definir los requisitos se utilizará una tabla que contendrá y mostrará la información perteneciente a cada requisito acorde al siguiente formato:

ID	Identificador	NOMBRE	Requisito de ejemplo
PRIORIDAD	Alta / Media / Baja	NECESIDAD	Esencial / Deseable / Baja
DEPEDENCIAS	Otro identificador	FECHA	DDMM/AAAA
DESCRIPCIÓN	Descripción del requisito. Descripción que permita al desarrollador tener una idea clara de la exigencia a cubrir que posteriormente se va a implementar.		

Tabla 3.3.1: Modelo

Los identificadores elegidos para los requisitos constan de dos partes claramente diferenciadas, separadas entre sí por un guión. La primera es un código formado por dos letras que describe el tipo de requisito que se trata, y la segunda parte es un número de tres cifras que sirve para secuenciar requisitos del mismo tipo, siendo el primero siempre el 001.

Los códigos de letras usados en el identificador de los requisitos pueden ser de los siguientes tipos:

- RF – Requisito funcional.
- RI – Requisito de interfaz.
- RR – Requisito de rendimiento.
- RD – Requisito de diseño.

3.3.1 REQUISITOS FUNCIONALES

ID	RF – 001	NOMBRE	Identificar usuario
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	13/10/2012
DESCRIPCIÓN	La aplicación permitirá el acceso a la misma por autenticación utilizando la identidad social del usuario en Twitter. La aplicación permitirá la identificación del usuario frente a Twitter mediante su Web, y recibirá la sesión de Twitter del usuario identificado.		

Tabla 3.3.1.1: Requisito funcional 1

ID	RF - 002	NOMBRE	Suscribir asignatura
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS	RF – 001	FECHA	13/10/2012
DESCRIPCIÓN	La aplicación permitirá la suscripción a cursos por parte de los alumnos. La aplicación recibirá el identificador del curso al que se desea suscribir y le incluirá en la lista de cursos suscritos.		

Tabla 3.3.1.2: Requisito funcional 2

ID	RF - 003	NOMBRE	Abandonar asignaturas
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS	RF – 001, RF - 002	FECHA	13/10/2012
DESCRIPCIÓN	La aplicación permitirá abandonar cursos, por parte de los alumnos, a las que se encuentren suscritos. La aplicación recibirá el identificador del curso que se desea eliminar de las suscripciones, y le eliminará de la lista de cursos suscritos.		

Tabla 3.3.1.3: Requisito funcional 3

ID	RF - 004	NOMBRE	Acceder a material docente
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS	RF – 001, RF – 002, RF - 008	FECHA	13/10/2012
DESCRIPCIÓN	La aplicación permitirá el acceso al contenido ofrecido en el tablón de los cursos a las que el alumno este suscrito, permitiendo la descarga de los contenidos existentes en el mismo. La aplicación mostrará el tablón de un curso suscrito por el alumno y que este haya previamente seleccionado.		

Tabla 3.3.1.4: Requisito funcional 4

ID	RF - 005	NOMBRE	Listar asignaturas
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS	RF – 001, RF - 002	FECHA	13/10/2012
DESCRIPCIÓN	La aplicación permitirá el listado de los cursos a los cuales el alumno se encuentra suscrito.		

Tabla 3.3.1.5: Requisito funcional 5

ID	RF - 006	NOMBRE	Eliminar alumnos
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS	RF – 001, RF - 002	FECHA	13/10/2012
DESCRIPCIÓN	La aplicación permitirá la eliminación de alumnos que se encuentran suscritos a un determinado curso. La aplicación mostrará un listado con los alumnos que actualmente se encuentran suscritos a un curso, y permitirá al profesor, la selección de uno de ellos para su borrado.		

Tabla 3.3.1.6: Requisito funcional 6

ID	RF - 007	NOMBRE	Listar alumnos
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS	RF – 001, RF - 002	FECHA	13/10/2012
DESCRIPCIÓN	La aplicación permitirá el listado de los alumnos que se encuentran suscritos a una determinada asignatura.		

Tabla 3.3.1.7: Requisito funcional 7

ID	RF - 008	NOMBRE	Mostrar tablón
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS	RF – 001	FECHA	19/10/2012
DESCRIPCIÓN	La aplicación permitirá la existencia de un tablón de contenidos por cada curso existente, que ofrecerá los contenidos docentes correspondientes a cada curso.		

Tabla 3.3.1.8: Requisito funcional 8

ID	RF - 009	NOMBRE	Modificar contenidos
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS	RF – 001, RF – 008, RF - 011	FECHA	19/10/2012
DESCRIPCIÓN	La aplicación permitirá la modificación de los contenidos existentes en el tablón de cada curso. La aplicación permitirá la modificación de los contenidos existentes en el tablón del curso, previa selección del contenido a modificar, actualizando el contenido del tablón.		

Tabla 3.3.1.9: Requisito funcional 9

ID	RF - 010	NOMBRE	Borrar contenidos
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS	RF – 001, RF – 008, RF - 011	FECHA	19/10/2012
DESCRIPCIÓN	La aplicación permitirá el borrado de los contenidos existentes en el tablón de un curso. La aplicación permitirá el borrado de los contenidos existentes en el tablón del curso, previa selección del contenido a borrar, actualizando el contenido del tablón.		

Tabla 3.3.1.10: Requisito funcional 10

ID	RF - 011	NOMBRE	Añadir contenidos
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS	RF – 001, RF - 008	FECHA	19/10/2012
DESCRIPCIÓN	La aplicación permitirá añadir contenidos nuevos al tablón del curso, siendo posible incluirlos en una sesión existente o creando una nueva, actualizando de este modo el contenido del tablón.		

Tabla 3.3.1.11: Requisito funcional 11

ID	RF - 012	NOMBRE	Compartir recurso
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS	RF – 001	FECHA	19/10/2012
DESCRIPCIÓN	La aplicación permitirá la compartición de recursos docentes mediante los contenidos ofrecidos en el tablón de cada curso.		

Tabla 3.3.1.12: Requisito funcional 12

ID	RF - 013	NOMBRE	Estructura del tablón
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS	RF - 008	FECHA	6/11/2012
DESCRIPCIÓN	La aplicación permitirá ofrecer contenidos mediante el tablón atendiendo a la siguiente estructura: <ul style="list-style-type: none">• El material ofrecido estará organizado por sesiones, siendo cada una de estas las clases impartidas en la asignatura.• Los contenidos ofrecidos en cada sesión se encontrarán organizados por orden de inserción.		

Tabla 3.3.1.13: Requisito funcional 13

3.3.2. INTERFACES EXTERNAS

En esta sección se describen los requisitos asociados a las interfaces de usuario asociadas a la aplicación *MoLe*.

ID	RI - 001	NOMBRE	Comunicación interna
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	5/12/2012
DESCRIPCIÓN	La aplicación será capaz de comunicar sus interfaces entre sí utilizando envoltura de tipo Parcelable. La información compartida entre actividades será como mínimo la sesión de <i>Twitter</i> .		

Tabla 3.3.2.1: Requisito de interfaz 1

ID	RI - 002	NOMBRE	Comunicación externa
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	5/12/2012
DESCRIPCIÓN	La aplicación será capaz de comunicar sus interfaces internas con el navegador para realizar la identificación del usuario. La información compartida entre el navegador y las actividades será se la sesión de <i>Twitter</i> .		

Tabla 3.3.2.2: Requisito de interfaz 2

ID	RI - 003	NOMBRE	Vista advertencia
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	5/12/2012
DESCRIPCIÓN	La aplicación mostrará un mensaje de advertencia al usuario, indicándole la política de tratamiento de datos existente en la misma, y un botón para que pueda aceptarla.		

Tabla 3.3.2.3: Requisito de interfaz 3

ID	RI - 004	NOMBRE	Vista asignaturas suscritas
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	5/12/2012
DESCRIPCIÓN	La aplicación mostrará un menú con las opciones de borrar y modificar curso, así como el listado de las asignaturas a las que el alumno se encuentra suscrito.		

Tabla 3.3.2.4: Requisito de interfaz 4

ID	RI - 005	NOMBRE	Vista suscribirse asignatura
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	5/12/2012
DESCRIPCIÓN	La aplicación mostrará un formulario para la suscripción a asignaturas compuesto por una caja de texto y un botón. El formato de nombre de la asignatura introducida en el formulario de suscripción será el mismo que habitualmente se utiliza en <i>Twitter</i> compuesto por @ seguido del nombre de la asignatura.		

Tabla 3.3.2.5: Requisito de interfaz 5

ID	RI - 006	NOMBRE	Vista abandonar asignatura
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	5/12/2012
DESCRIPCIÓN	La aplicación mostrará un listado de las asignaturas a las cuales el alumno se encuentra suscrito y un botón identificativo que permitirá el borrado de las mismas.		

Tabla 3.3.2.6: Requisito de interfaz 6

ID	RI - 007	NOMBRE	Vista tablón alumno
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	5/12/2012
DESCRIPCIÓN	La aplicación mostrará la información perteneciente a cada tablón en forma de lista, diferenciando mediante colores y tamaños el material descargable y las cabeceras referentes a las clases.		

Tabla 3.3.2.7: Requisito de interfaz 7

ID	RI - 008	NOMBRE	Vista tablón asignatura
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	5/12/2012
DESCRIPCIÓN	La aplicación mostrará un menú con las opciones de ver lista de alumnos, modificar contenido de tablón, añadir contenido a tabón y refrescar tablón. Además mostrará la información contenida en el tablón de la asignatura en forma de lista, diferenciando mediante colores y tamaños el material descargable y las cabeceras referentes a las clases.		

Tabla 3.3.2.8: Requisito de interfaz 8

ID	RI - 009	NOMBRE	Vista listado alumnos
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	5/12/2012
DESCRIPCIÓN	La aplicación mostrará un menú con las opciones de borrar alumno y refrescar alumnos, además de una lista compuesta por los alumnos que se encuentran suscritos a dicha asignatura.		

Tabla 3.3.2.9: Requisito de interfaz 9

ID	RI - 010	NOMBRE	Vista borrar alumnos
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	5/12/2012
DESCRIPCIÓN	La aplicación mostrará un listado con los alumnos suscritos a una asignatura y un botón representativo con forma de aspa roja para su borrado.		

Tabla 3.3.2.10: Requisito de interfaz 10

ID	RI - 011	NOMBRE	Vista modificar contenido
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	5/12/2012
DESCRIPCIÓN	La aplicación mostrará la información contenida en el tablón de la asignatura identificando los contenidos susceptibles a modificación o borrado con signos de admiración encerrados entre corchetes (“[!]”).		

Tabla 3.3.2.11: Requisito de interfaz 11

ID	RI - 012	NOMBRE	Vista formulario modificación
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	11/12/2012
DESCRIPCIÓN	La aplicación mostrará un formulario para la modificación del contenido del tablón compuesto por dos cajas de texto, la primera para establecer el nombre del recurso y la segunda para la ruta del recurso a compartir, un botón para adjuntar recursos y un botón para aceptar los cambios realizados.		

Tabla 3.3.2.12: Requisito de interfaz 12

ID	RI - 013	NOMBRE	Vista explorador ficheros
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	17/12/2012
DESCRIPCIÓN	La aplicación mostrará un explorador de ficheros para seleccionar el recurso docente que se desea compartir.		

Tabla 3.3.2.13: Requisito de interfaz 13

ID	RI - 014	NOMBRE	Vista añadir recurso
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	5/12/2012
DESCRIPCIÓN	La aplicación mostrará la información contenida en el tablón de la asignatura identificando los días de clase con signos de admiración encerrados entre corchetes (“[!]”) para indicar la posibilidad de añadir nuevos recursos a ese día, y dos nuevas opciones, situadas en la parte superior e inferior de la lista, que habilitan la posibilidad de crear un nuevo día de clase y añadir contenidos al mismo.		

Tabla 3.3.2.14: Requisito de interfaz 14

ID	RI - 015	NOMBRE	Vista formulario añadir día
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	11/12/2012
DESCRIPCIÓN	La aplicación mostrará un formulario para la creación de un nuevo día de clase dentro del tablón. El formulario estará compuesto por tres cajas de texto, la primera para establecer la fecha de la clase, la segunda para el nombre del recurso a compartir, y la tercera para la ruta del recurso a compartir, un botón para adjuntar recursos y un botón para aceptar los cambios realizados.		

Tabla 3.3.2.15: Requisito de interfaz 15

3.3.3 REQUISITOS DE RENDIMIENTO

ID	RR - 001	NOMBRE	Tiempo máximo de respuesta
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	13/12/2012
DESCRIPCIÓN	Se establece un tiempo máximo de respuesta de 5 segundos para cualquier consulta asociada a un recurso ajeno al terminal. En caso el caso de superar los 5 segundos una consulta la aplicación mostrará un formulario emergente indicando que la aplicación no responde, permitiendo la opción de esperar a que responda o cancelándola, finalizando el proceso de la misma.		

Tabla 3.3.3.1: Requisito de rendimiento 1

3.3.4 RESTRICCIONES DE DISEÑO

La única restricción de diseño impuesta por el cliente ha sido relacionada con la versión del sistema operativo sobre el que se ejecutará la aplicación, que se estableció que fuera *Android 4.0.3*.

ID	RD - 001	NOMBRE	Versión del sistema operativo
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	13/12/2012
DESCRIPCIÓN	La aplicación deberá ser compatible con dispositivos Android que posean una versión 4.0.3 “Ice cream sandwich” o superior.		

Tabla 3.3.4.1: Requisito de diseño 1

ID	RD - 002	NOMBRE	Versión del SDK de Java
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	13/12/2012
DESCRIPCIÓN	El lenguaje de programación a utilizar será Java y el nivel de compilación será el 1.6 dado que es el que se establece por defecto para este tipo de proyectos software.		

Tabla 3.3.4.2: Requisito de diseño 2

ID	RD - 003	NOMBRE	API de comunicación
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	21/01/2013
DESCRIPCIÓN	La aplicación utilizará una API definida para la comunicación entre la aplicación y el servidor de contenidos.		

Tabla 3.3.4.3: Requisito de diseño 3

ID	RD - 004	NOMBRE	API de Twitter
PRIORIDAD	Alta	NECESIDAD	Esencial
DEPEDENCIAS		FECHA	21/01/2013
DESCRIPCIÓN	La aplicación utilizará las librerías de Twitter4j en su versión 3.0.3 de la API de Twitter para la comunicación con Twitter.		

Tabla 3.3.4.4: Requisito de diseño 4

3.3.5 ATRIBUTOS DEL SISTEMA

La aplicación será estable, siendo necesario para ello que se controlen todos los posibles errores que se pudieran generar en el funcionamiento de la misma. Además el usuario deberá ser notificado de posibles malos funcionamientos, con la intención de que, en la medida de lo posible, pueda solucionar los posibles conflictos.

La aplicación deberá poder ser actualizada y mantenida por parte de otro desarrollador sin dificultades, atendiendo a la documentación disponible para la misma.

La aplicación no será portable en el estado de desarrollo actual de la misma, siendo Android la única plataforma sobre la que funcionará.

La aplicación deberá ser capaz de responder ante usos maliciosos de la misma. La seguridad de la aplicación se encuentra localizada en el proceso de autenticación con Twitter, al cual delegamos la seguridad en este aspecto, confiando en sus facultades y buena praxis, y a la comunicación con el servidor, el cual posee una capa de seguridad integrada para proteger de ataques de red convencionales.

CAPÍTULO 4 : DISEÑO DE LA APLICACIÓN

4.1 PROTOTIPO	47
4.1.1 MOLE ALUMNOS.....	47
4.1.2 MOLE PROFESORES.....	52
4.2 ALTERNATIVAS DE DISEÑO	61
4.2.1 DISEÑO DE LA APLICACIÓN	64
4.3 DIAGRAMA DE CLASES.....	64
4.3.1 DIAGRAMA DE CLASES MOLE ALUMNOS.....	65
4.3.2 DIAGRAMA DE CLASES MOLE PROFESORES.....	68
4.4 DIAGRAMAS DE SECUENCIA	74

En este capítulo se presenta el *prototipo* que se realizó en los primeros análisis del problema, el cual ha sido modificado en cada iteración del proceso de desarrollo hasta dar lugar al estado actual de la aplicación. El objetivo de estos prototipos es aproximar gráficamente la idea de aplicación que se posee, con la intención de compartirla con el cliente y que así pueda ver aproximadamente como quedará la aplicación. De este modo es más sencillo pulir detalles, dado que esta fase se encuentra en un estado anterior al desarrollo de la misma.

Posteriormente se describirán las alternativas de diseño surgidas en el proceso de desarrollo, aportando las ventajas y desventajas que cada una posea, y explicando cuál finalmente de ellas se ha escogido.

Para concluir se ofrecerá un diagrama de componentes, de tal modo que el lector pueda comprender de una manera más rápida el diseño llevado a cabo en este trabajo.

4.1 PROTOTIPO

En esta sección se muestra el prototipo realizado para la aplicación entre los meses de Noviembre y Diciembre del año 2012.

Como preámbulo hay que destacar que la aplicación MoLe está dividida en dos aplicaciones diferentes, una para profesores y otra para alumnos. Por esta razón en este apartado se mostrará el prototipo de realizado para cada una de ellas de forma independiente, tal y como se muestra en la figura 4.1.1, la cual indica su separación y cómo operan frente a sistemas externos como si fueran una sola.

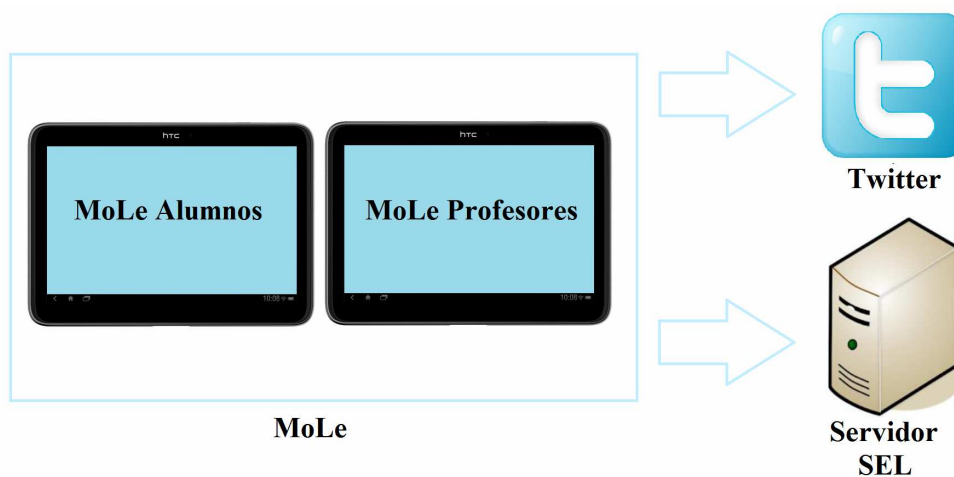


Figura 4.1.1: Prototipo

4.1.1 MOLE ALUMNOS

En la figura 4.1.1.1 se muestra la pantalla inicial de la aplicación. En ella se le ofrece un aviso al usuario, indicándole la política de tratamiento de datos que la aplicación posee, y las instrucciones necesarias para continuar con la ejecución de la aplicación.

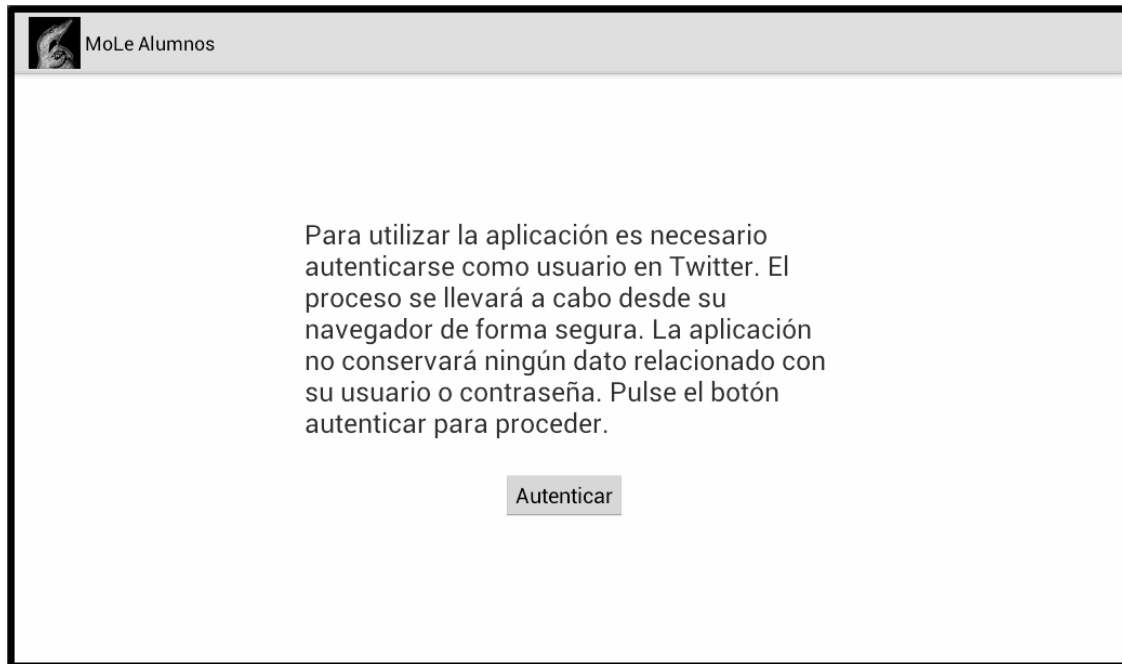


Figura 4.1.1.1: Vista inicio

Tras pulsar el botón de autenticar el usuario será redireccionado al navegador, y cargada la página Web de Twitter, donde el usuario deberá introducir sus datos de inicio de sesión y aceptar los términos de uso. Tras aceptar los términos se le redireccionará nuevamente a la aplicación. En la figura 4.1.2 se muestra la vista principal del alumno a la cual le es llevado tras aceptar los términos de Twitter en el navegador.



Figura 4.1.1.2: Vista principal

En la figura 4.1.2 se muestra la vista principal del alumno, en la cual podrá ver la lista de cursos a las que se encuentra suscrito y podrá suscribirse a nuevos o darse de baja de en los que actualmente esté suscrito. Desde la vista general y utilizando el menú de la parte superior de la interfaz el usuario puede suscribirse a un nuevo curso, mediante el icono de la lupa, o darse de baja de un curso al que está suscrito mediante el icono del lápiz. En la figura 4.1.3 se muestra la interfaz referente a la suscripción a cursos.

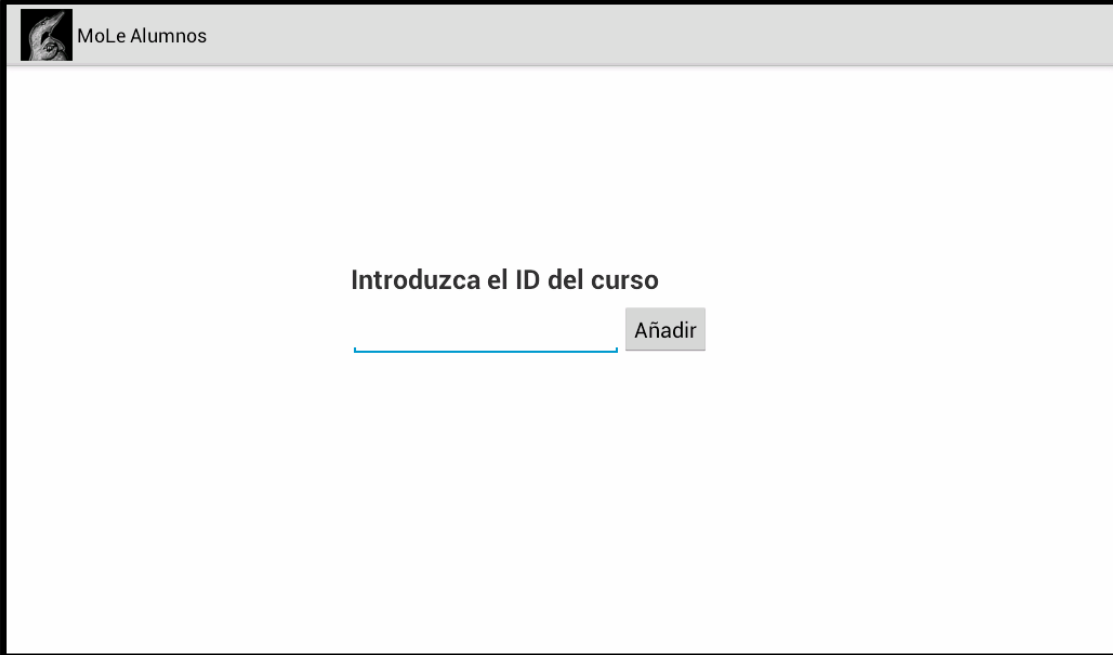
The image shows a web interface for 'MoLe Alumnos'. At the top left, there is a small profile picture icon and the text 'MoLe Alumnos'. The main area of the page is white and contains the text 'Introduzca el ID del curso' in a bold, dark font. Below this text is a light blue rectangular input field. To the right of the input field is a grey button with the text 'Añadir' in white. The entire interface is enclosed in a black rectangular border.

Figura 4.1.1.3: Suscripción curso

En esta figura el usuario podrá suscribirse a un curso al que actualmente no se encuentre suscrito y que exista como usuario de Twitter. Para ello es necesario que introduzca el nombre del curso, precedido de “@” y pulse el botón añadir.

Desde la figura 4.1.2 pulsando el botón del lápiz se accede a la figura 4.1.4 referente a darse de baja en cursos suscritos.

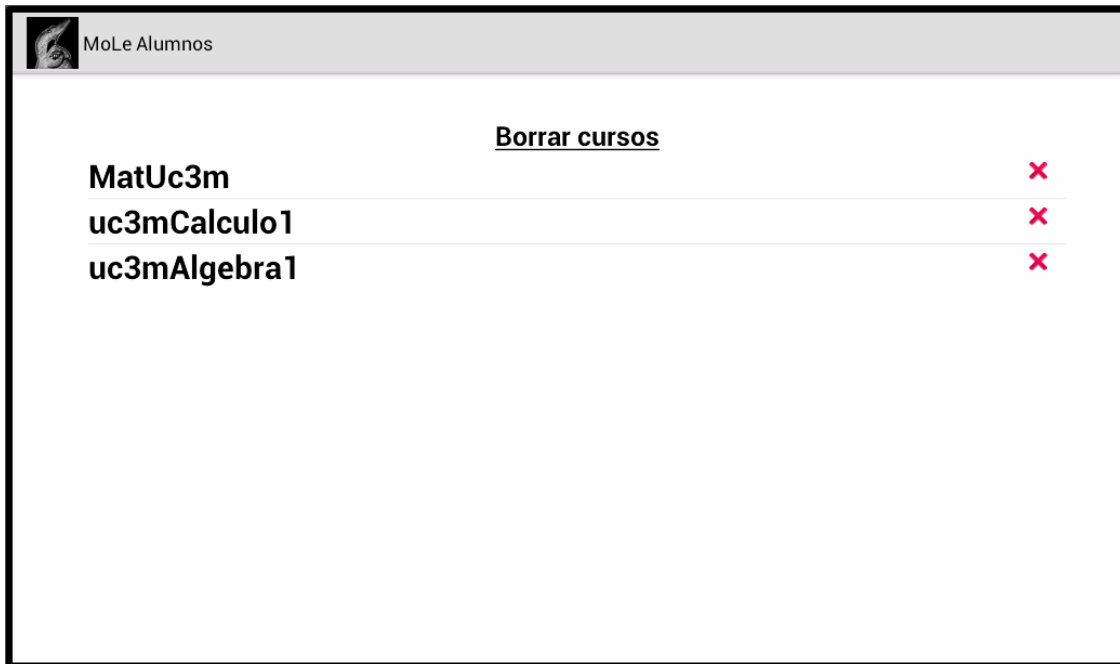


Figura 4.1.1.4: Baja curso

En esta figura se ofrece el listado de los cursos en los que el alumno se encuentra suscrito con la novedad de un icono en forma de aspa roja que indica que el curso es susceptible a su borrado. Para proceder al borrado del mismo es necesario pinchar sobre el curso deseado, mostrando la figura 4.1.5 que se muestra a continuación.

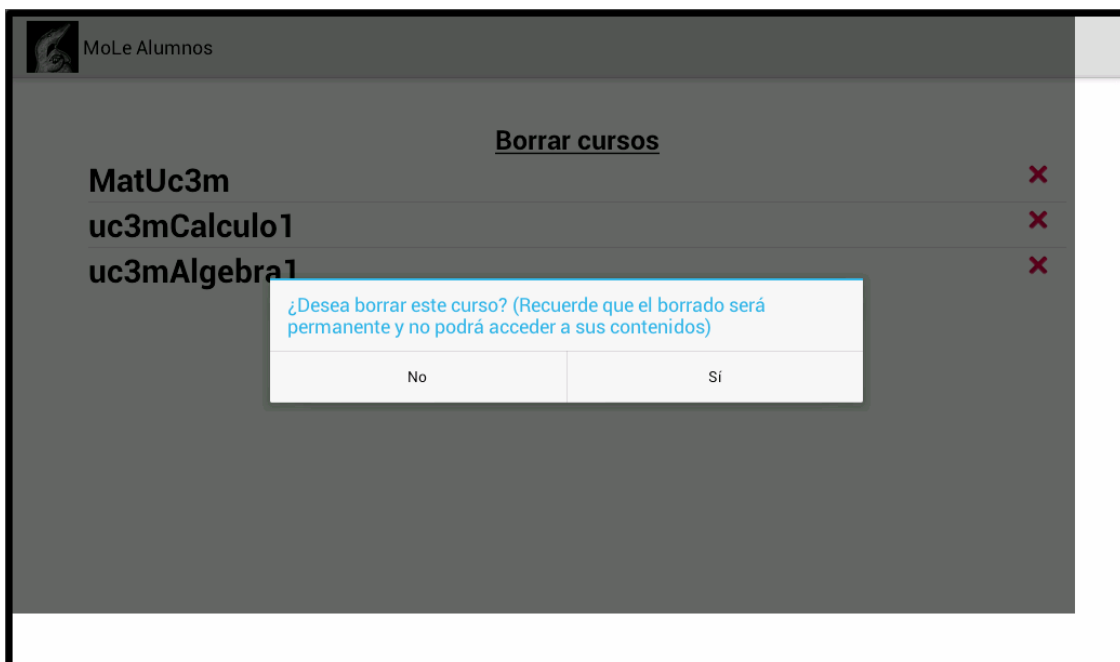


Figura 4.1.1.5: Baja emergente

Atendiendo a la figura 4.1.5 si el usuario pulsara en la opción “Sí”, tal y como indica la advertencia el curso seleccionado quedaría borrado, habiéndose dado de baja del mismo. Por otro lado si el usuario pulsara la opción “No” la aplicación no realizará ninguna modificación sobre los cursos suscritos.

Por ultimo, para concluir MoLe Alumnos, desde la figura 4.1.2 de todas los cursos los que el alumno se encuentra suscrito y que se le muestran en forma de listado el alumno puede seleccionar uno de ellos y acceder a la figura 4.1.6 en la que se muestra el tablón de dicho curso. Dicha figura se muestra a continuación.

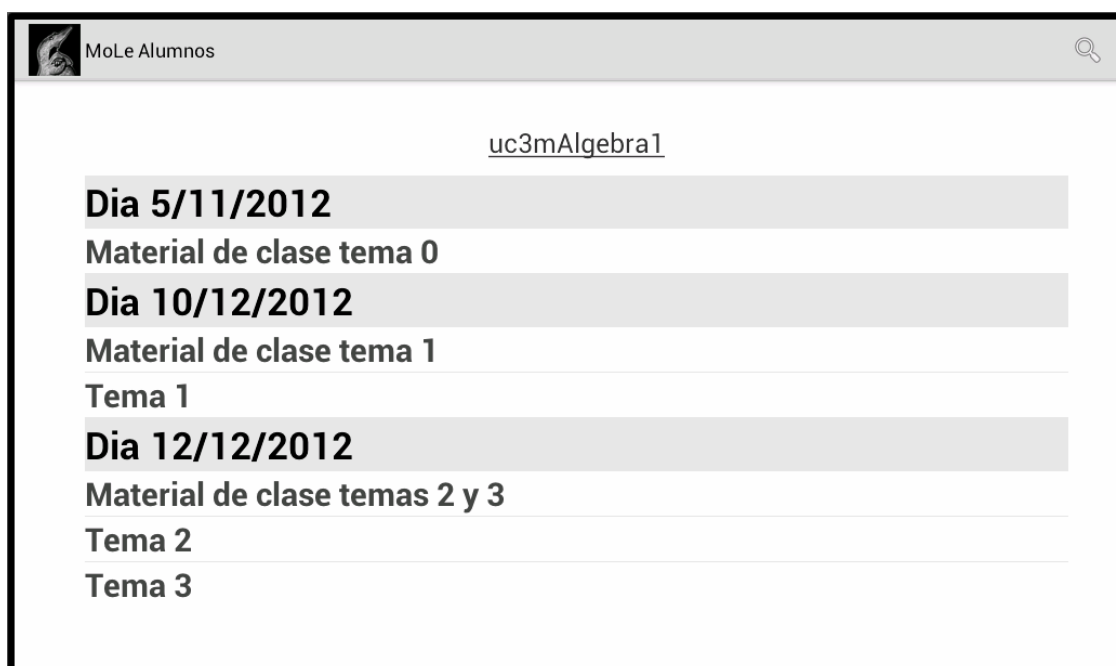


Figura 4.1.1.6: Tablón alumno

En esta figura se muestra el tablón perteneciente al curso seleccionado por el alumno. En este se ofrecen los contenidos compartidos por el profesor a través del curso, para que los alumnos que se encuentran suscritos a el puedan acceder a los mismos. El tablón tiene dos tipos de datos bien diferenciados, las sesiones, caracterizadas por una fecha y estar en negrita, y por otro lado los contenidos ofrecidos en cada sesión, siendo los contenidos añadidos por el profesor a cada sesión y que el alumno puede descargar. Para la descarga de cualquier contenido tan solo es necesario pinchar en el recurso deseado y la descarga se inicia.

4.1.2 MOLE PROFESORES

En la figura 4.1.2.1 se muestra la pantalla inicial de la aplicación. En ella se le ofrece un aviso al usuario, indicándole la política de tratamiento de datos que la aplicación posee, y las instrucciones necesarias para continuar con la ejecución de la aplicación.

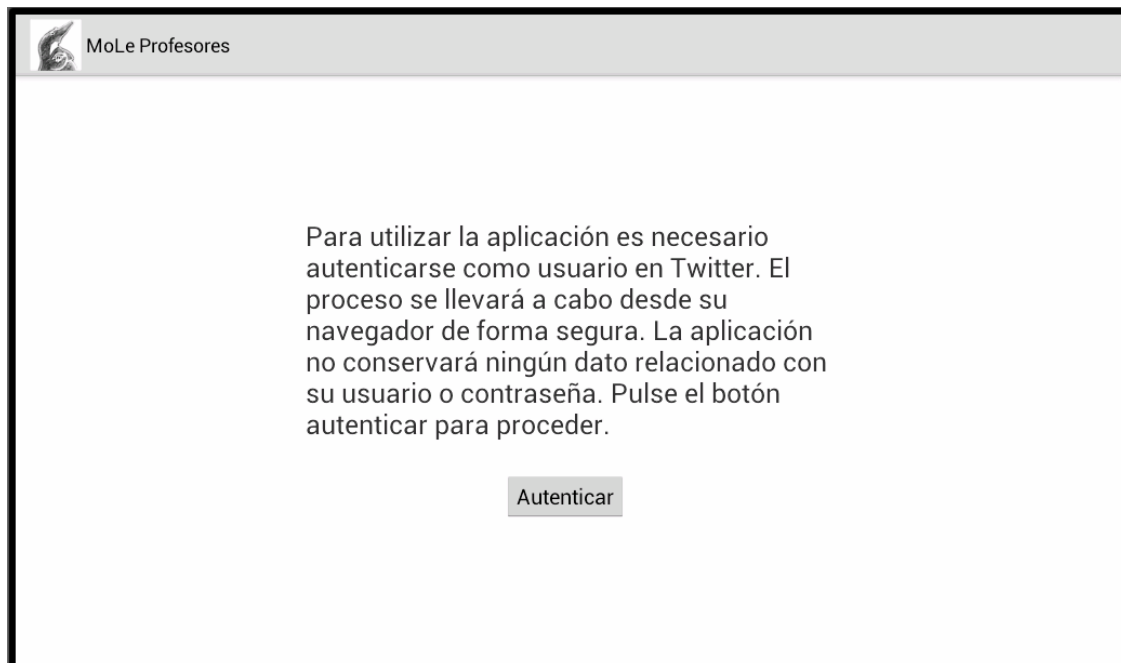
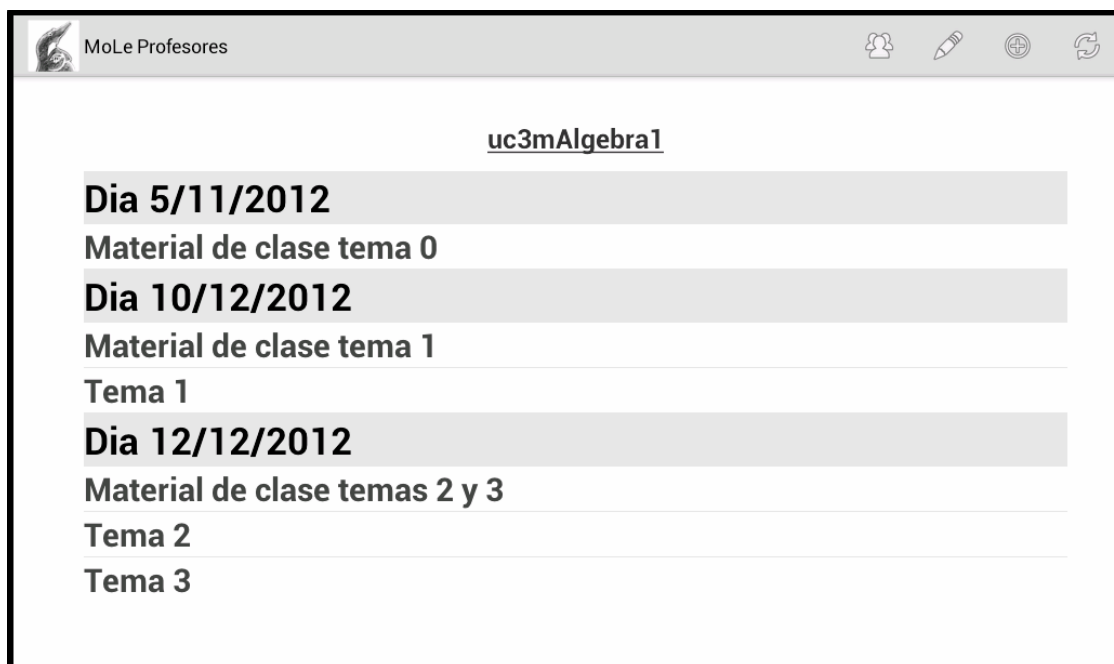


Figura 4.1.2.1: Vista inicio

Tras pulsar el botón de autenticar el usuario será redirigido al navegador, y cargada la página Web de Twitter, donde el usuario deberá introducir sus datos de inicio de sesión y aceptar los términos de uso. Tras aceptar los términos se le redirigirá nuevamente a la aplicación. En la figura 4.1.2.2 se muestra la vista principal del curso a la cual le es llevado tras aceptar los términos de Twitter en el navegador.

**Figura 4.1.2.2: Vista principal**

La figura 4.1.2.2 muestra el tablón del curso que actualmente el profesor administra a través de la aplicación. En ella se muestra el contenido que actualmente se ofrece a través de su tablón, y una serie de opciones para administrar tanto los contenidos que se ofrecen como los alumnos que acceden a ellos. Desde esta figura el profesor puede acceder a la información referente a los alumnos suscritos a su curso mediante el icono de los “hombrecitos” que se encuentra en la barra de opciones situada en la parte superior de la interfaz. Dicha información se muestra en la figura 4.1.2.3 que se ofrece a continuación.

**Figura 4.1.2.3: Vista alumnos**

En esta figura se listan los alumnos que actualmente se encuentran suscritos a la asignatura y se permite desde las opciones ofrecidas en la parte superior de la interfaz administrarlos. De este modo a través del icono del “lápiz” se accede a la figura 4.1.2.4 que permite al profesor el borrado de alumnos suscritos a su asignatura. Dicha figura se muestra a continuación.



Figura 4.1.2.4: Borrado alumnos

En esta figura se muestran los alumnos suscritos al curso, en forma de lista, tal y como se mostraban en la figura 4.1.2.3, con la salvedad que ahora se muestra en cada uno de ellos un icono en forma de aspa roja, que indica que los alumnos son susceptibles a borrado. Para proceder al borrado de alumnos es necesario pinchar sobre el alumno deseado y se mostrará el contenido ofrecido por la figura 4.1.2.5 que se muestra a continuación.

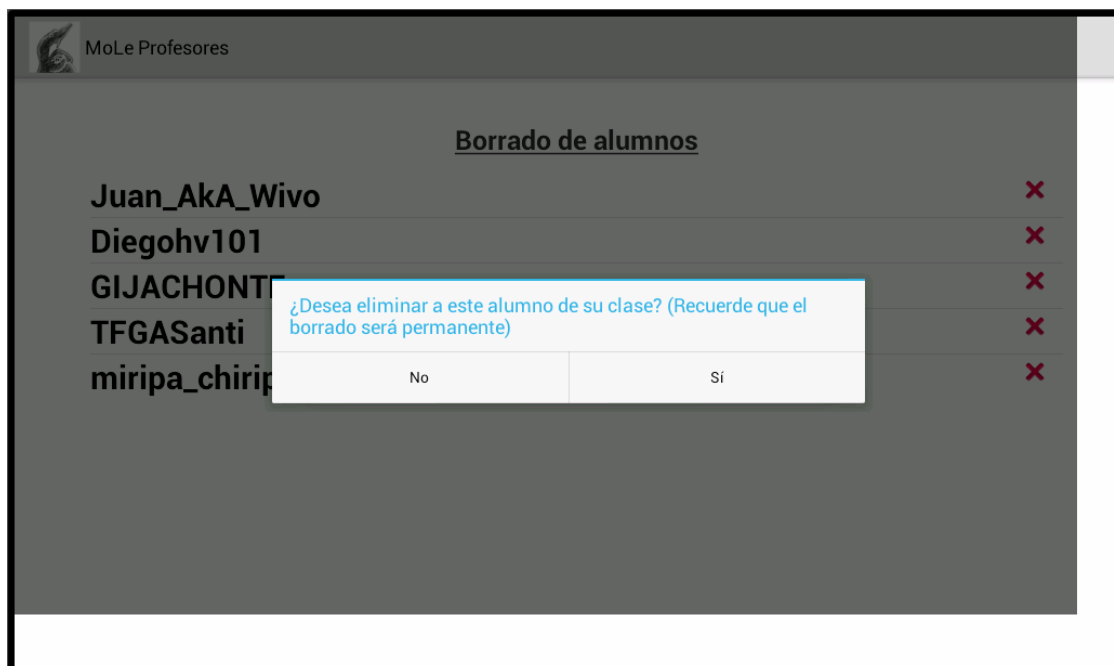


Figura 4.1.2.5: Borrado alumno emergente

En esta figura se muestra la ventana emergente desplegada que aparece cuando se selecciona un alumno para su borrado desde la figura 4.1.2.4. En ella se muestra una advertencia en la que se indica que el borrado del alumno será permanente, y se le ofrecen dos opciones “Si” y “No”. Si el usuario pulsara la opción “Si” el alumno quedaría excluido del curso. Por el contrario si el usuario pulsara “No” no se produciría ninguna modificación en el listado de alumnos suscritos.

Desde la figura 4.1.2.2 mediante el icono del “lápiz” se le permite al profesor administrar los contenidos existentes en el tablón de su curso, y modificar contenidos ofrecidos en el mismo. En la figura 4.1.2.6 se muestra la interfaz existente para la modificación de contenidos, la cual se muestra a continuación.

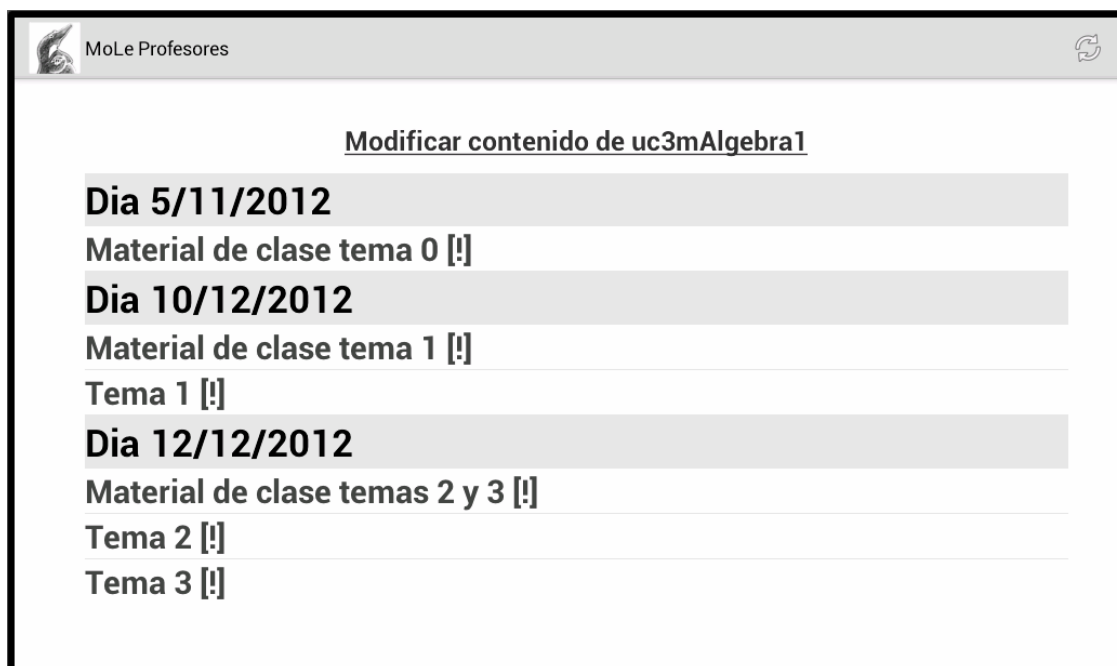


Figura 4.1.2.6: Modificar curso

En esta figura se muestra el contenido ofrecido en el tablón del curso susceptible a modificación, siendo este contenido sensible el indicado con signos de admiración encerrados entre corchetes “[!]”. El profesor podrá modificar cualquiera de los contenidos indicados con este símbolo pinchando sobre el y seleccionando una de las opciones ofrecidas en la figura 4.1.2.7 que se muestra a continuación.

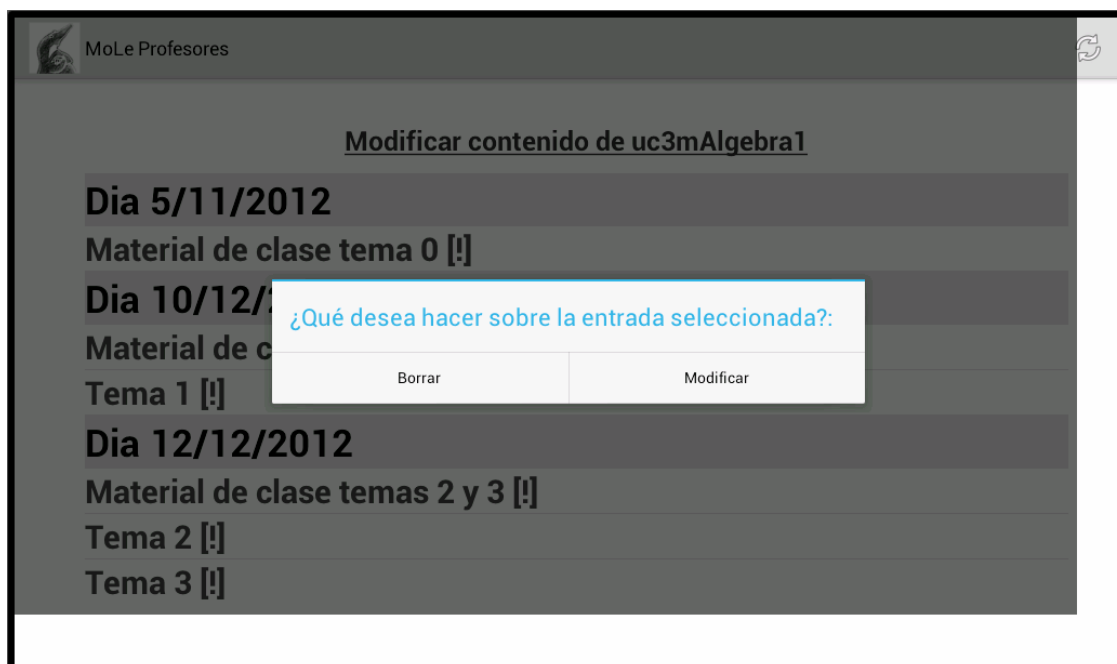
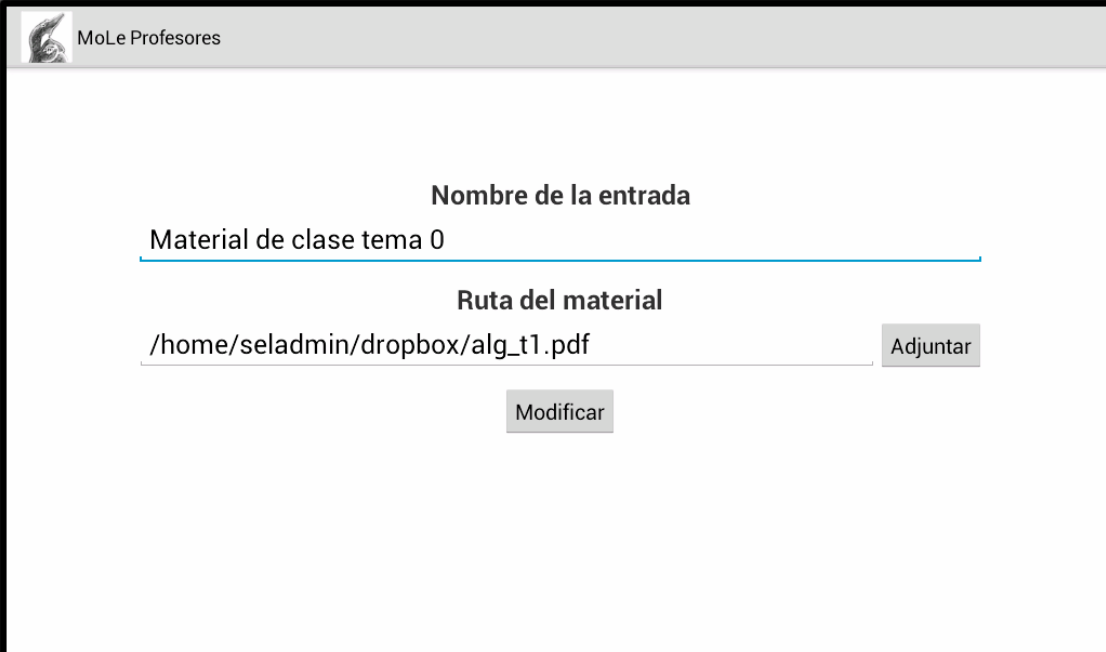


Figura 4.1.2.7: Modificar curso emergente

En esta figura se muestra la ventana emergente resultado de pinchar en un contenido susceptible a modificación. La ventana ofrece dos opciones, “borrar” y “modificar”. Si el profesor escogiese la opción “borrar” el recurso seleccionado del tablón sería borrado. Si este fuera el único recurso ofrecido en esa sesión, además del recurso, también será borrado el identificador de la sesión mostrado en el tablón. Por otro lado, si el profesor escogiese la opción modificar será dirigido a la figura 4.1.2.8 en la cual se le permitirá la modificación de los datos pertenecientes al contenido seleccionado. Dicha figura se muestra a continuación.



MoLe Profesores

Nombre de la entrada

Material de clase tema 0

Ruta del material

/home/seladmin/dropbox/alg_t1.pdf

Adjuntar

Modificar

Figura 4.1.2.8: Formulario modificación

En esta figura se muestra el contenido actual ofrecido en la entrada del tablón seleccionada por el profesor para su modificación, mostrando los datos que actualmente contiene, y posibilitando al profesor su modificación, tanto en nombre a mostrar como en recurso a compartir. Para modificar el nombre del recurso es necesario alterar el contenido de la caja de texto “Nombre de la entrada” al gusto, y por otro lado para modificar el material a compartir es necesario pinchar en el botón “Adjuntar” para seleccionar el nuevo archivo a compartir. Dicho botón ofrece la siguiente figura 4.1.2.9 cuando se selecciona, la cual se muestra a continuación.

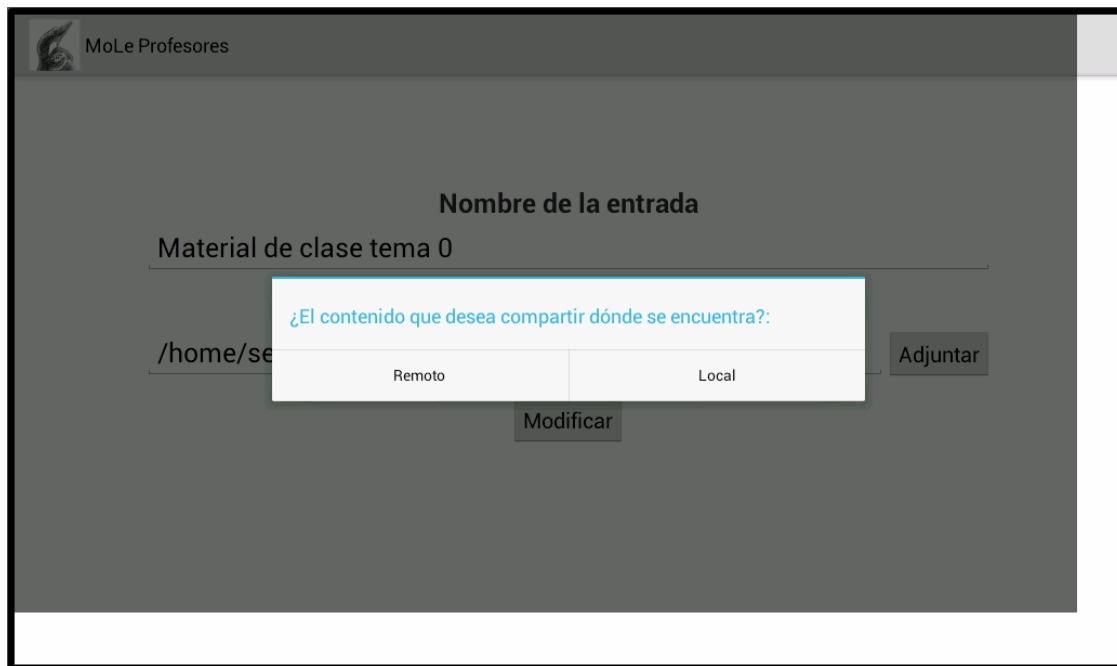


Figura 4.1.2.9: Formulario modificación adjuntar

En esta figura se muestra la ventana emergente producto de pinchar sobre el botón “Adjuntar” de la figura 4.1.2.9. En ella se muestran las dos opciones ofrecidas a la hora de compartir contenidos. Las opciones ofrecidas permiten la compartición de material desde el propio terminal móvil, o bien a través de un recurso remoto. Las figuras 4.1.2.10 y 4.1.2.11 muestran las dos posibilidades de compartición, las cuales se muestran a continuación.



Figura 4.1.2.10: Explorador archivos local

En esta figura se muestra el explorador de ficheros del terminal móvil. En este se le permite al profesor seleccionar un contenido existente en su terminal para su compartición con el alumnado. Para ello deberá navegar entre los directorios de su terminal, pintados con el color azul, hasta encontrar el recurso que desea compartir, pintado en blanco, sobre el cual deberá pinchar para su selección. A continuación se redirigirá a la figura 4.1.2.8 con el fichero a compartir modificado, sustituyendo el antiguo por el nuevo que acaba de seleccionar. Para concluir el profesor deberá pinchar en el botón “Modificar” para volver a la figura 4.1.2.7 donde los cambios realizados se muestran al profesor.

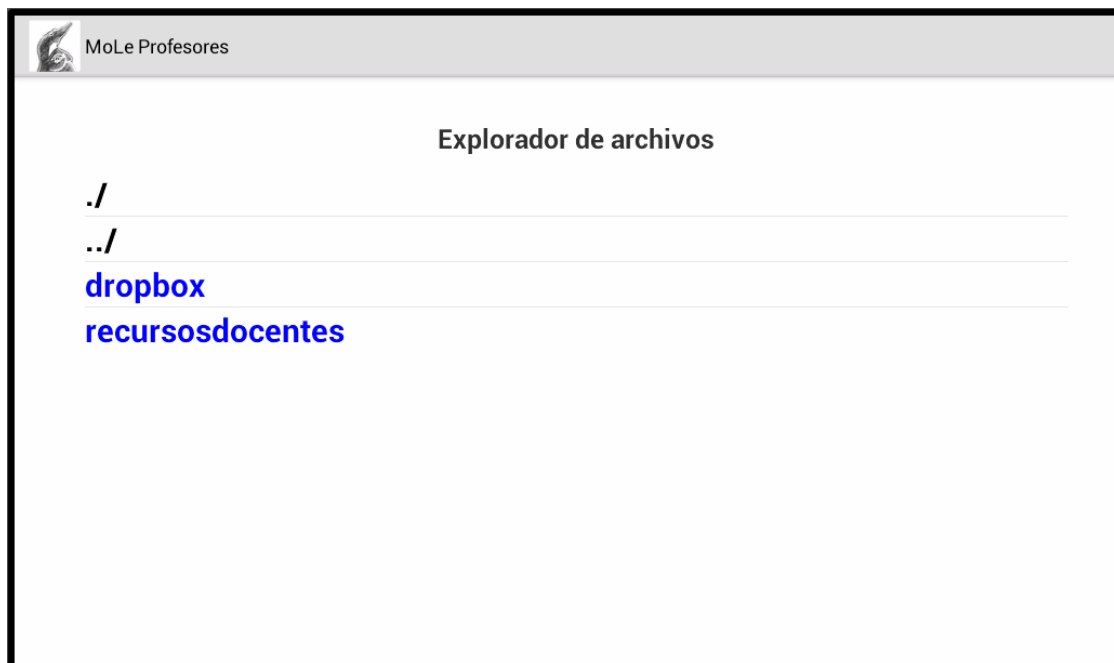
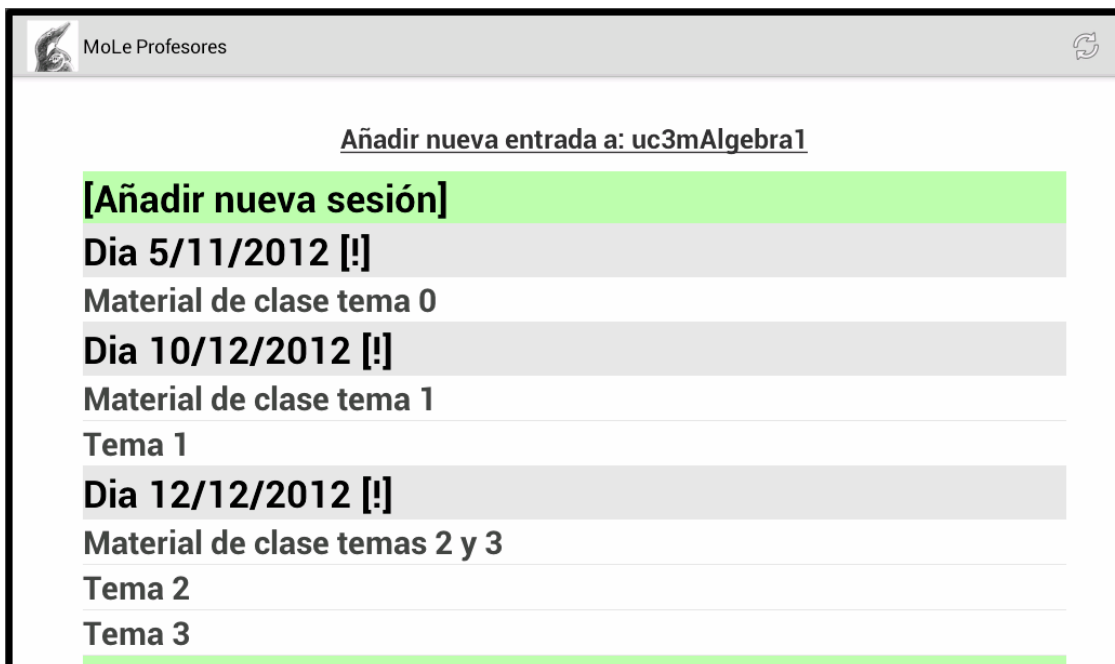


Figura 4.1.2.11: Explorador archivos remoto

En esta figura se muestra el explorador de ficheros remotos. En este se le permite al profesor seleccionar un contenido que no se encuentra en su terminal para su compartición con el alumnado. Para ello deberá navegar entre los directorios del recurso remoto, pintados con el color azul, hasta encontrar el recurso que desea compartir, pintado en blanco, sobre el cual deberá pinchar para su selección. A continuación se redirigirá a la figura 4.1.2.8 con el fichero a compartir modificado, sustituyendo el antiguo por el nuevo que acaba de seleccionar. Para concluir el profesor deberá pinchar en el botón “Modificar” para volver a la figura 4.1.2.7 donde los cambios realizados se muestran al profesor.

Desde la figura 4.1.2.2 el profesor podrá añadir nuevos contenidos al tablón de su curso. Para ello deberá pinchar sobre el icono en forma de “+” situado dentro de un círculo que se encuentra en las opciones de la parte superior de la interfaz para ser llevado a la figura 4.1.2.12 que se muestra a continuación.



Añadir nueva entrada a: uc3mAlgebra1

[Añadir nueva sesión]

Día 5/11/2012 [!]
Material de clase tema 0

Día 10/12/2012 [!]
Material de clase tema 1

Tema 1

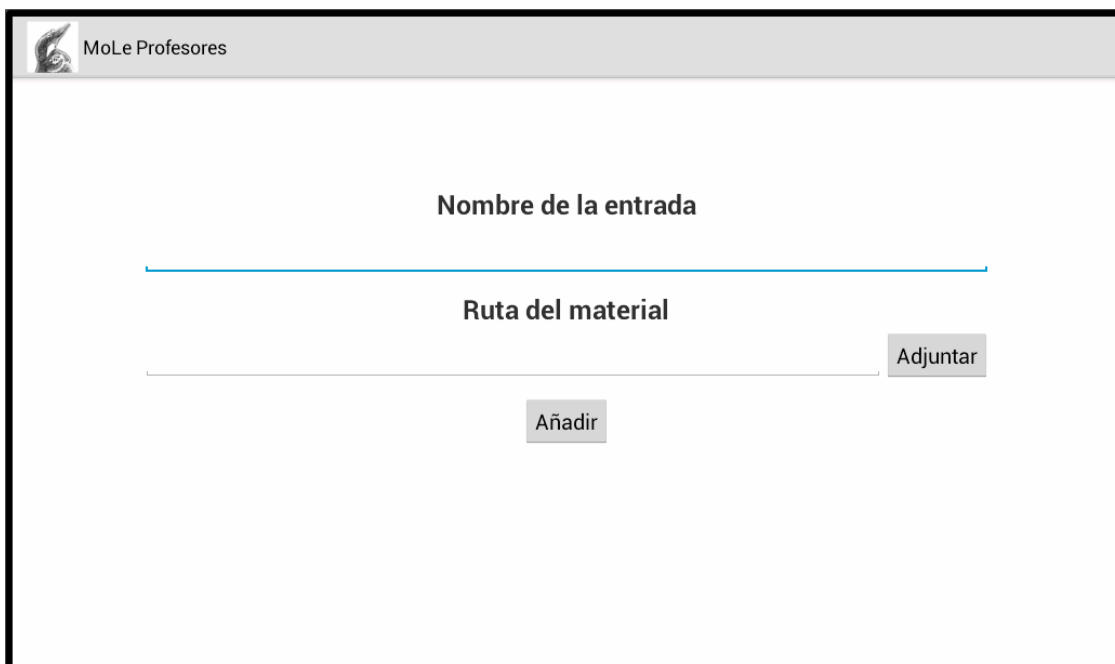
Día 12/12/2012 [!]
Material de clase temas 2 y 3

Tema 2

Tema 3

Figura 4.1.2.12: Añadir nueva entrada

En esta figura se muestra el contenido existente en el tablón del curso, y se le permite añadir nuevos recursos utilizando dos vías. La primera es añadir un contenido en una sesión ya existente, indicada con el signo “[!]”, la cual se muestra en la figura 4.1.2.13. La segunda es añadir un contenido en una sesión nueva, seleccionando la opción “[Añadir nueva sesión]” que se muestra en color verde en la parte superior e inferior de la interfaz, la cual se muestra en la figura 4.1.2.14. A continuación se ofrecen las figuras de ambas vías anteriormente mencionadas.



Nombre de la entrada

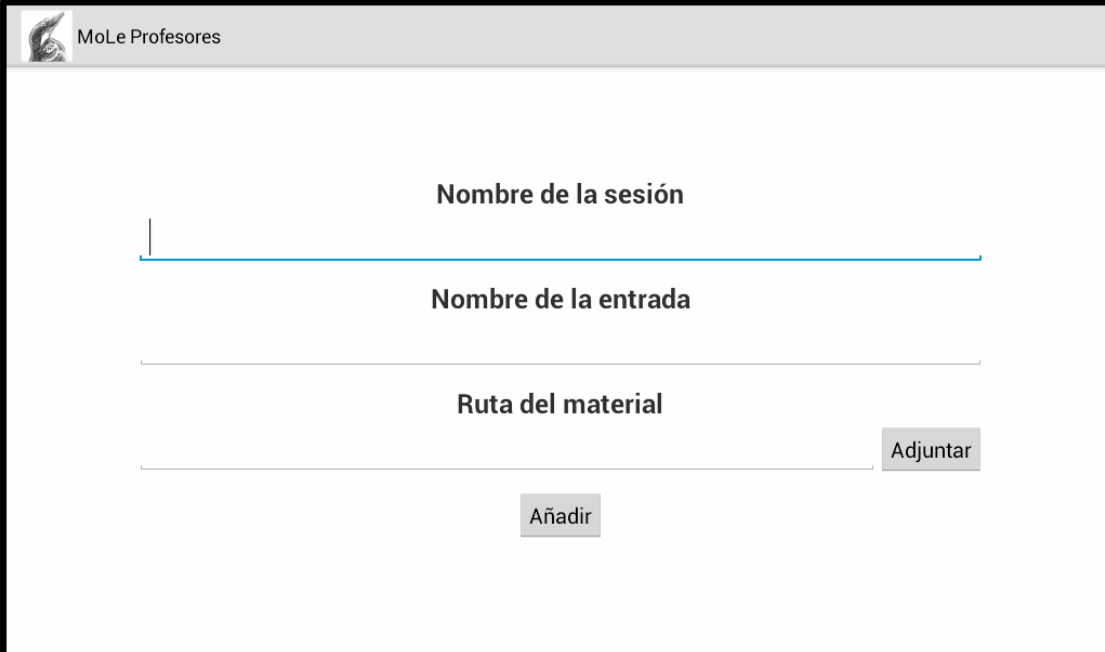
Ruta del material

Adjuntar

Añadir

Figura 4.1.2.13: Añadir entrada a sesión

En esta figura se muestra el formulario ofrecido para añadir contenidos en una sesión ya existente. Para ello el profesor ha de rellenar el nombre del material que se mostrará en el tablón del curso y deberá seleccionar un recurso para compartir, mediante el botón adjuntar, procediendo al mismo mecanismo de la figura 4.1.2.9 anteriormente mencionado. Una vez completos todos los campos se deberá pulsar el botón “Añadir” para proceder a incluir dicho recurso en la sesión seleccionada previamente por el profesor. A continuación la aplicación volverá a la figura 4.1.2.12 mostrando el material ofrecido de forma actualizada.



The screenshot shows a web interface for 'MoLe Profesores'. It features a form with three input fields: 'Nombre de la sesión', 'Nombre de la entrada', and 'Ruta del material'. To the right of the 'Ruta del material' field is a button labeled 'Adjuntar'. Below the 'Ruta del material' field is a button labeled 'Añadir'. The form is enclosed in a black border.

Figura 4.1.2.14: Añadir sesión nueva

En esta figura se muestra el formulario ofrecido para añadir contenidos en una nueva sesión. Para ello el profesor ha de rellenar el nombre de la nueva sesión, el nombre del material que se mostrará en el tablón del curso y deberá seleccionar un recurso para compartir, mediante el botón adjuntar, procediendo al mismo mecanismo de la figura 4.1.2.9 anteriormente mencionado. Una vez completos todos los campos se deberá pulsar el botón “Añadir” para proceder a incluir dicha sesión y recurso en el tablón del curso. A continuación la aplicación volverá a la figura 4.1.2.12 mostrando el material ofrecido de forma actualizada.

4.2 ALTERNATIVAS DE DISEÑO

En esta sección se explican brevemente las alternativas de diseño obtenidas en la fase de diseño de la aplicación, y de ellas cual ha sido seleccionada como el diseño a implantar en la aplicación.

Servicio Web: Al principio del proyecto se tuvo que decidir que tipo de lenguaje se iba a utilizar para manejar los datos pertenecientes a los tabloneros de los cursos, contenido en el servidor. Las alternativas fueron XML y JSON. Pese a la simplicidad inicial que ofrecía JSON, y el menor flujo de información entre el cliente y el servidor, la opción de XML resultó más atractiva por la complejidad y disparidad en la estructura del tablón, el cual era más sencillo de manejar si se recibía como fichero XML, organizado con múltiples etiquetas, para su posterior tratamiento y manejo. Pese a que el fichero XML introduzca múltiples cabeceras que incrementan en gran medida el volumen de la información a enviar y recibir no supone un problema estructural grave en la velocidad de la aplicación, dado que la información contenida en éstos ficheros no suele ser muy pesada, y el tiempo incrementado por dicho espacio es casi imperceptible de cara al usuario del cliente.

Estructura de los datos en XML: Los ficheros XML manejados para contener toda la información relativa al contenido del tablón de cada curso necesitaban de una estructura definida, que permitieran por un lado su modificación y recuperación de la manera más sencilla posible, y por otro que poseyera el mínimo volumen de cabeceras posible, con la intención contener el tamaño del fichero en el mínimo espacio posible. Para ello se desarrollaron tres tipos de etiquetas, cada una compuesta por 3 caracteres, que diferencian la sesión usando la etiqueta “<s>”, el nombre de cada sesión utilizando la etiqueta “<h>”, y el nombre de cada recurso usando la etiqueta “”.

El formato utilizado en los XML de los tabloneros es el siguiente:

<s>

<h>Día 5/11/2012</h>

Material de clase tema 0/home/seladmin/dropbox/alg_t1.pdf

</s>

<h>Día 10/12/2012</h>

Material de clase tema 1/mnt/sdcard/TFG/MaterialClase.rar

Tema 1/mnt/sdcard/TFG/alg_t1.pdf

</s>

<h>Día 12/12/2012</h>

Material de clase temas 2 y 3/mnt/sdcard/TFG/MaterialClase2.rar

Tema 2/mnt/sdcard/TFG/alg_t2.pdf

Tema 3/mnt/sdcard/TFG/alg_t3.pdf

Unificación de contenidos: Uno de los primeros debates en el diseño de la aplicación fue si esta debía ser ofrecida en bloque o dividida en partes, es decir, una aplicación conteniendo toda la funcionalidad, tanto para profesores como alumnos, convirtiéndola en una aplicación universal para ambos, o bien dividir su funcionalidad en

dos aplicaciones independientes, una para profesores y otra para alumnos. Sopesando las ventajas e inconvenientes que ambas opciones poseían se llegó a la conclusión de que separarlas era lo más acertado, dado que, por un lado, un profesor difícilmente va a ser al mismo tiempo alumno, y viceversa, anulando una parte de la funcionalidad en base al rol del usuario que la vaya a utilizar. Por otro lado dividiéndola también se consigue que el instalador de la aplicación sea más ligero, que ocupe menos espacio en memoria, y que consuma menos recursos del terminal. Como tercer punto a favor de separar la funcionalidad se encuentra la dificultad por parte de alumnos con malas intenciones de hacerse pasar por profesores, suplantando su identidad. Con la división no se le impide que lo haga, pero tiene un añadido de trabajo más, descargar e instalar la aplicación para profesores.

Por todas estas razones la aplicación se ha decidido dividir en dos partes, ejerciendo como aplicaciones independientes “MoLe Profesores” y “MoLe Alumnos”.

Estructuración de contenidos en tablón: La forma en la que se dispone la información en el tablón del curso también estuvo sujeta a debate, habiendo dos opciones bien diferenciadas para la distribución del mismo. La primera fue añadir contenidos a granel, siguiendo la misma disposición que tiene *AulaGlobal2*, definiendo la organización interna de los mismos el propio orden que se llevo a cabo para su inserción. La segunda opción esbozada fue establecer identificadores para cada sesión de clase, y vincular contenidos a dicho identificador, de tal modo que los contenidos ofrecidos queden organizados en base a las sesiones existentes. En esta alternativa los recursos docentes compartidos en el tablón se insertan en la primera posición de la sesión correspondiente, haciéndolos más visibles de cara al usuario, en contraposición a ocupar el ultimo lugar como se proponía en la otra alternativa. Por motivos de orden y limpieza a la hora de mostrar los contenidos del tablón se escogió como alternativa de diseño en este aspecto la segunda opción.

Método de identificación con *Twitter*: La forma de identificación manejada por la aplicación utilizando la identidad social del usuario en *Twitter* tuvo dos variantes de diseño. La primera fue embeber un *WebView* dentro de la actividad de identificación de la aplicación, y contener allí el formulario de acceso a *Twitter*, así como sus identificaciones. La segunda opción fue la redirección del usuario al navegador y obligarle a que acceda a *Twitter* desde su página Web, sin que la aplicación interceda en este proceso. Ambas opciones de diseño resultaban correctas, y pese a que la primera opción resultara mas cómoda de cara al usuario se escogió finalmente la segunda opción para la identificación por dos razones. La primera la desvinculación de la aplicación con respecto a los datos del usuario en *Twitter*, evitando que ésta participe en el proceso de introducir usuario y contraseña conseguimos que el usuario confíe más en la aplicación, dado que la identificación se realiza frente a *Twitter*, tal y como se hace siempre. La segunda es el espacio, que impedía mostrar toda la información desplegada en la página de *Twitter* destinada a la identificación. Era un requisito básico que ambos formatos fueran idénticos, ya que si difieren en algún aspecto el usuario puede pensar que la página no es la oficial, sino un clon con fines maliciosos.

4.2.1 DISEÑO DE LA APLICACIÓN

La aplicación se comunicará con dos sistemas independientes a la misma, por un lado Twitter, el cual proveerá de los servicios básicos de acceso, altas y bajas, y por otro lado al servidor, que almacenará los contenidos ofrecidos en los tableros de los cursos. Como extra la aplicación se comunicará con un tercer sistema, siendo este no obligatorio para el funcionamiento de la misma, un segundo servidor que ejerza de repositorio de contenidos para el profesor.

La estructura de la aplicación por lo tanto queda definida de la siguiente forma:

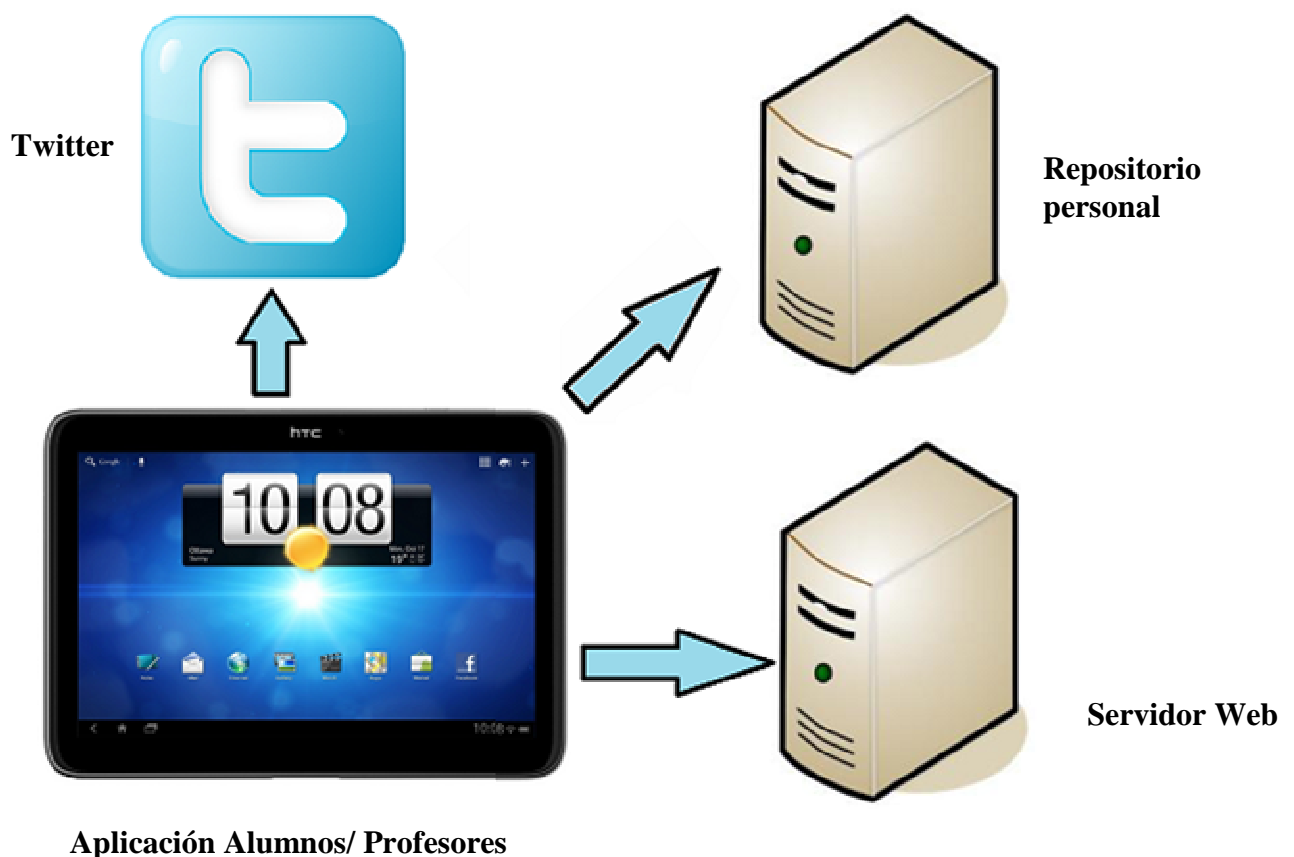


Figura 4.2.1: Diseño de la aplicación

4.3 DIAGRAMA DE CLASES

Tras valorar y escoger entre todas las alternativas de diseño se ha realizado un diseño de la aplicación por medio de diagramas. Se ha realizado un diagrama de clases que muestra las clases existentes en la aplicación, acorde al diseño establecido, los atributos y métodos que poseen, y por ultimo las relaciones básicas entre ellas. Dichas relaciones solo serán representadas si se focalizan entre elementos pertenecientes al diseño, ya que en Java una gran mayoría de las clases implementan o heredan de otras, por lo que el diagrama de clases sería intratable y poco representativo.

Dado que este trabajo consta de dos aplicaciones independientes se diferenciarán las clases pertenecientes a MoLe Alumnos y MoLe Profesores indicando las clases pertenecientes a cada aplicación.

4.3.1 DIAGRAMA DE CLASES MOLE ALUMNOS

La clase FTPClientInterface es un API de comunicación desarrollado para estandarizar la comunicación entre la aplicación y el servidor. Para ello ha sido necesaria su implementación en MyFTPClient. Dicha clase implementa todos los métodos que el interfaz tiene definidos y aporta la funcionalidad necesaria para la comunicación en ambos sentidos, entre la aplicación y el servidor. Figura 4.3.1.1

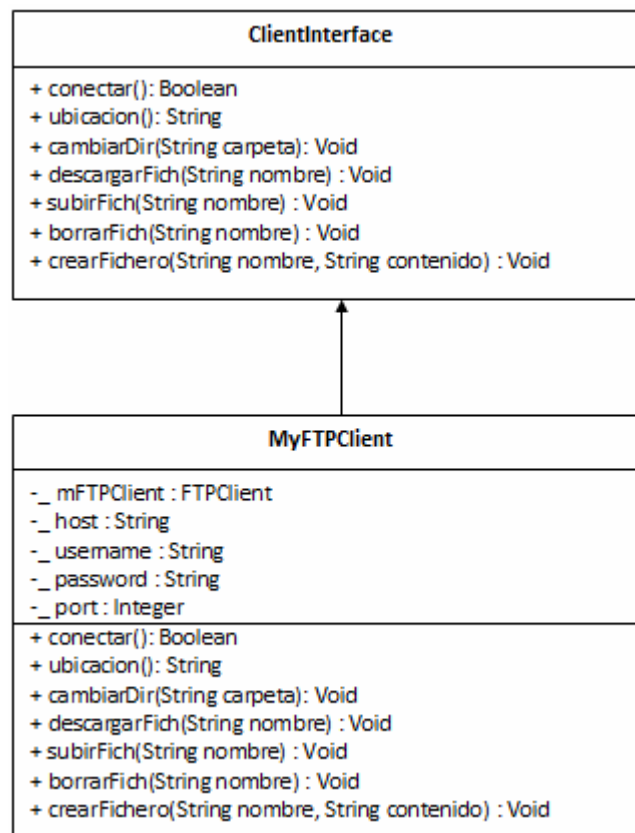


Figura 4.3.1.1: diagrama 1

La clase MyParcelable será usada para el encapsulamiento de objetos de tipo Twitter en objetos de tipo Parcelable, con la intención de poder enviar dichos objetos entre actividades a través de sus intent. Dicho proceso no era posible llevarlo a cabo sin utilizar este tipo de encapsulamiento, por lo que fue necesaria su implementación. Figura 4.3.1.2.

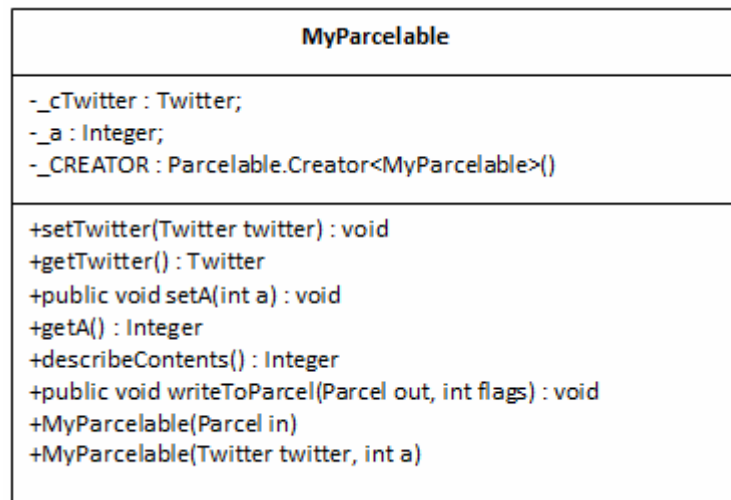


Figura 4.3.1.2: diagrama 2

La clase ParserXML será la encargada de transformar los ficheros XML recibidos en datos usables por la aplicación. De igual modo esta se encargará de realizar el efecto inverso, es decir, de transformar los datos usados por la aplicación en un fichero XML correctamente estructurado para su posterior subida al servidor utilizando los métodos de la clase FTP. Figura 4.3.1.3.

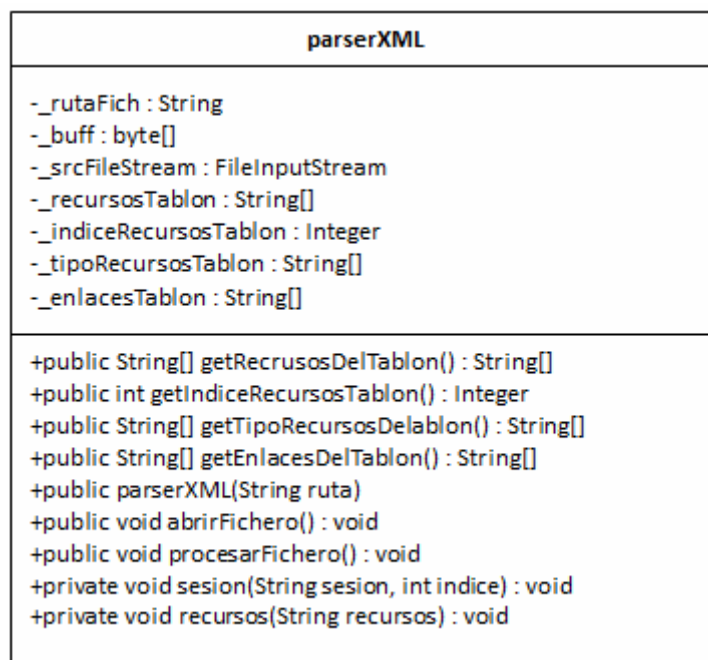


Figura 4.3.1.3: diagrama 3

A continuación se describirán todas las clases referentes a actividades de la aplicación MoLe Alumnos.

En la figura 4.3.1.4 se muestran las clases TFGActivity, TFGBorrarCurso, y TFGAniadirCurso. Estas clases son actividades de la aplicación, por lo que todas están asociadas a interfaces de usuario. La clase TFGActivity es la encargada de establecer la comunicación con Twitter y recoger el objeto generado en la autenticación. Las clases TFGBorrarCurso y TFGAniadirCurso son las encargadas de listar los cursos existentes para que el usuario seleccione cual desea borrar, y mostrar un formulario para suscribirse a cursos respectivamente.

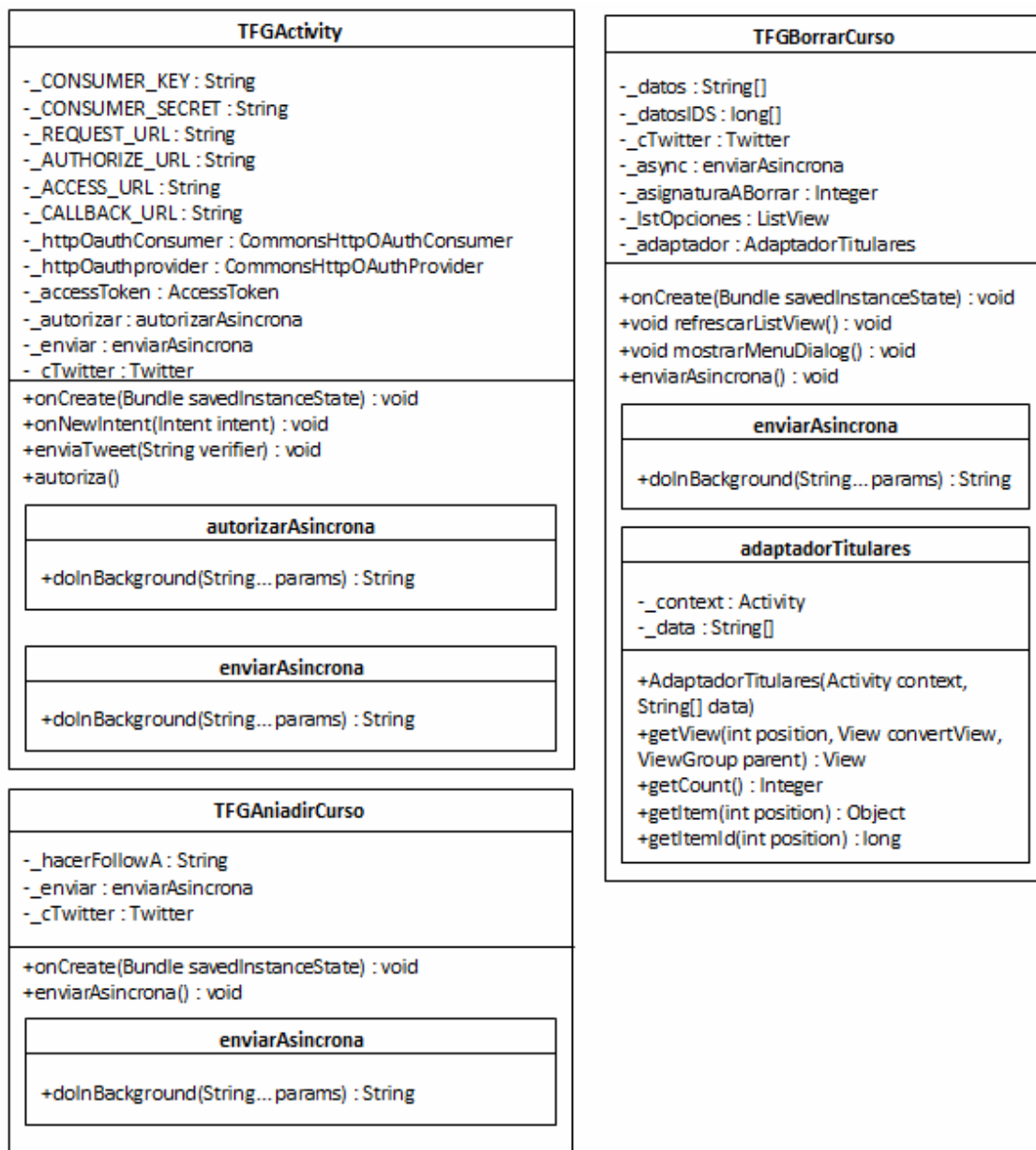


Figura 4.3.1.4: diagrama 4

En la figura 4.3.1.5 se muestran las clases TFGCurso y TFGHall. Dichas clases, al igual que las anteriormente mencionadas, también son actividades, y tienen un respaldo en forma de interfaz de usuario con el que el usuario interactúa. La funcionalidad de TFGCurso es desplegar el contenido del tablón de un curso cuando el alumno lo selecciona, y la clase TFGHall es la encargada de listar los cursos a los que el usuario se encuentra suscrito.

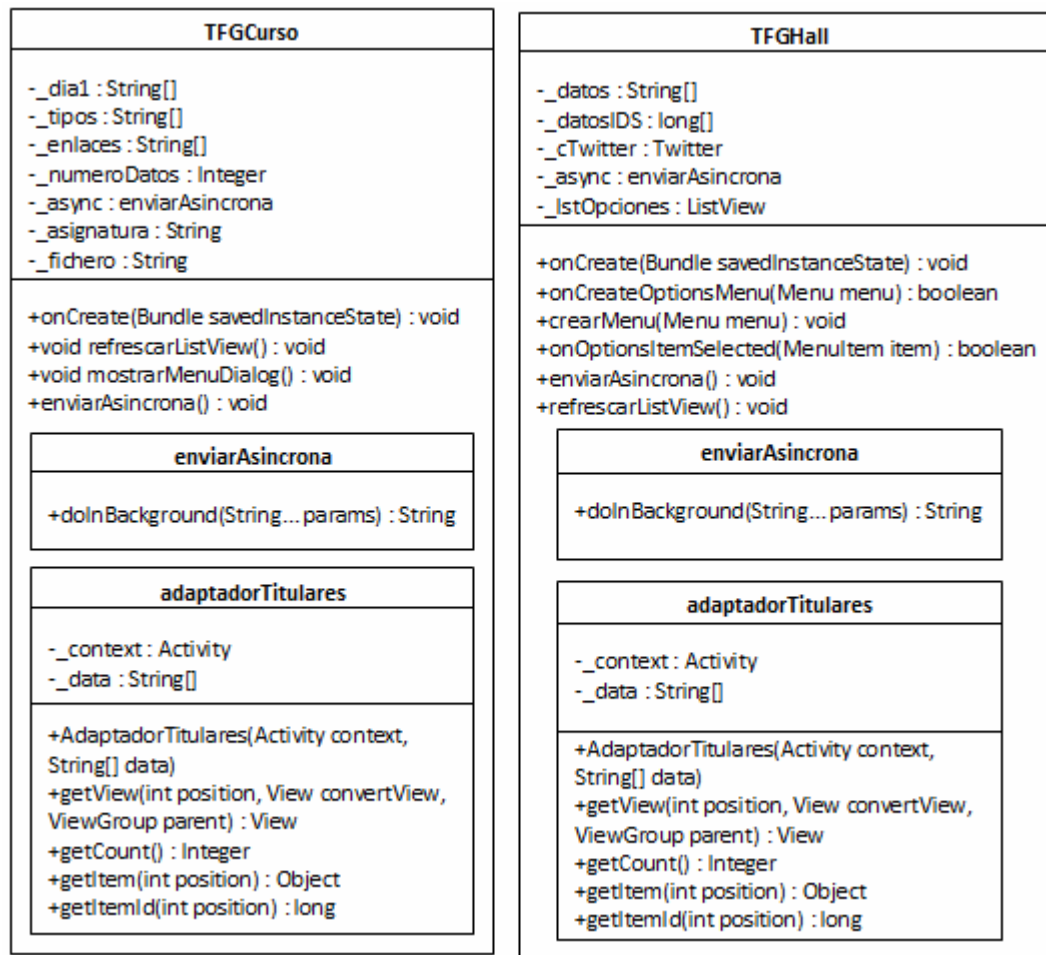


Figura 4.3.1.5: diagrama 5

4.3.2 DIAGRAMA DE CLASES MOLE PROFESORES

En la aplicación MoLe Profesores al igual que en MoLe Alumnos están las clases FTPClientInterface, MyFTPClient, MyParcelable y parserXML, todas ellas idénticas en forma y funcionalidad a las mostradas en la sección 4.3.1 en las figuras 4.3.1.1, 4.3.1.2, y 4.3.1.3.

En la figura 4.3.2.1 se muestran las clases TFGAlumnos y TFGAlumnosPendientes. Dichas clases son actividades, y tienen como objetivo por un lado el listado de los alumnos suscritos al curso y el listado de alumnos que intentan suscribirse al curso respectivamente. Ambas clases poseen una clase que genera un hilo de ejecución asíncrono para realizar las conexiones de red.

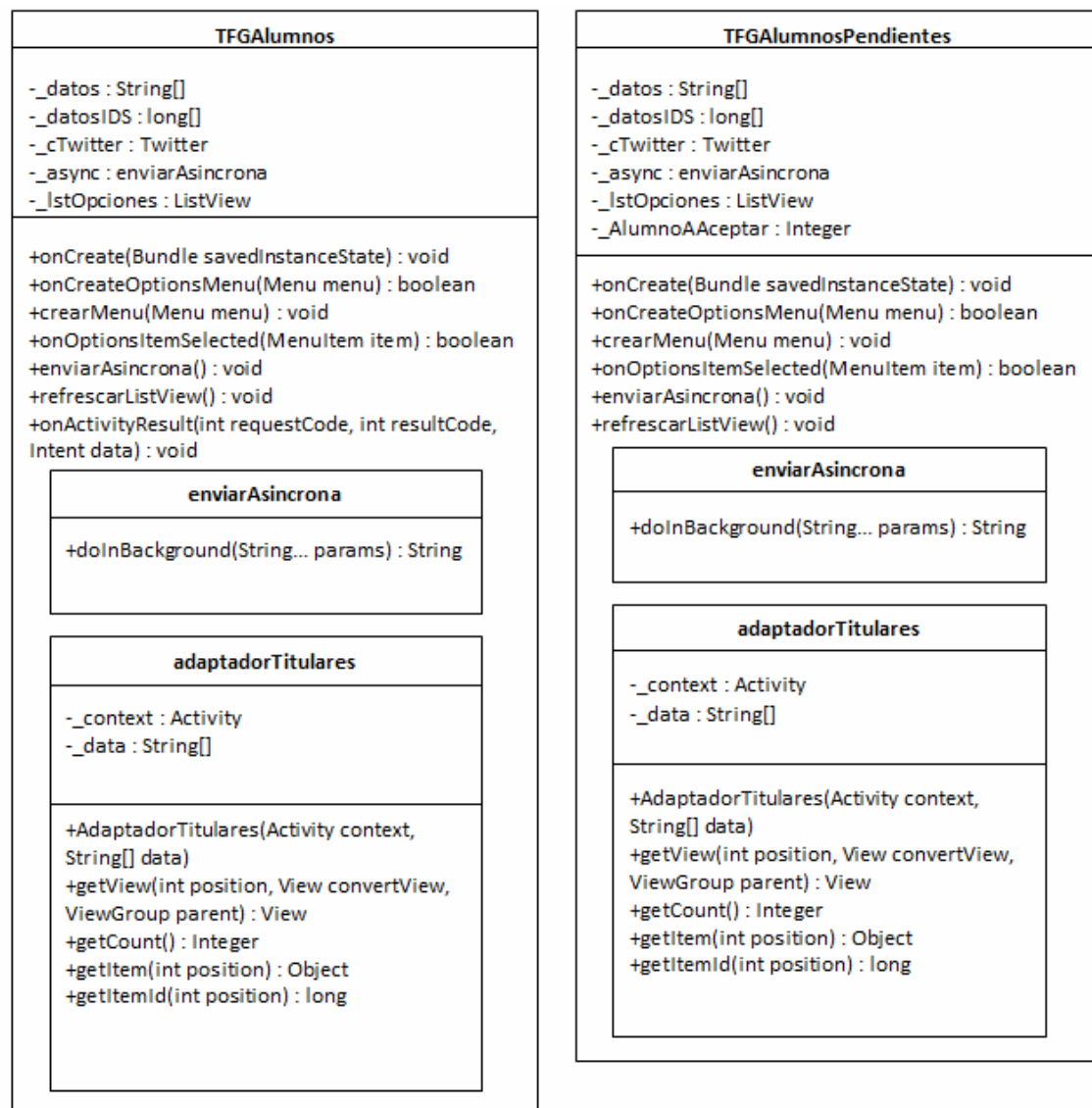


Figura 4.3.2.1: diagrama 1

En la figura 4.3.2.2 se muestran las clases TFGAniadirEntrada y TFGAniadirEntradaDiaExistente. Dichas clases son actividades y tienen como propósito incluir nuevos contenidos dentro del tablón del curso. TFGAniadirEntrada esta orientada a mostrar los contenidos del tablón sobre los que se pueden añadir nuevos recursos, y por otro lado TFGAniadirEntradaDiaExistente tiene como propósito incluir nuevos recursos en sesiones ya existentes en el tablón. Tal y como se describió en las actividades anteriores estas también poseen un hilo de ejecución asíncrono para realizar conexiones de red.

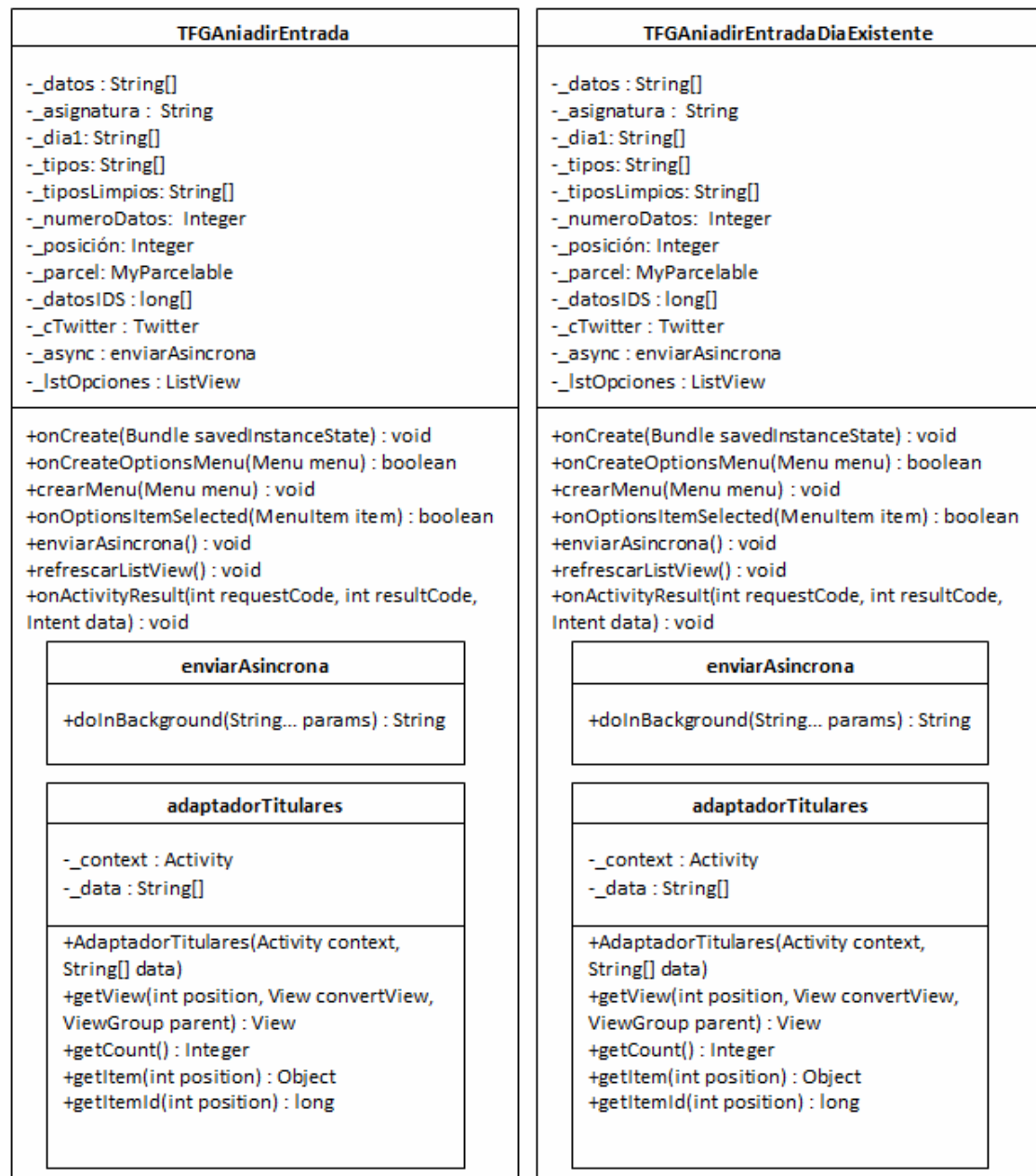


Figura 4.3.2.2: diagrama 2

En la figura 4.3.2.3 se muestran las clases TFGAniadirEntradaDiaNoExistente y TFGBorrarAlumno. Ambas clases son actividades y su propósito es, por un lado, el de TFGAniadirEntradaDiaNoExistente incluir nuevos recursos en el tablón del curso creando una nueva sesión para los mismos, y TFGBorrarAlumno permite la exclusión de alumnos suscritos al curso, eliminándolos de la lista de clase respectivamente. Ambas clases poseen una clase dentro de sí, que crea y maneja un hilo de ejecución asíncrono encargado de las conexiones de red.

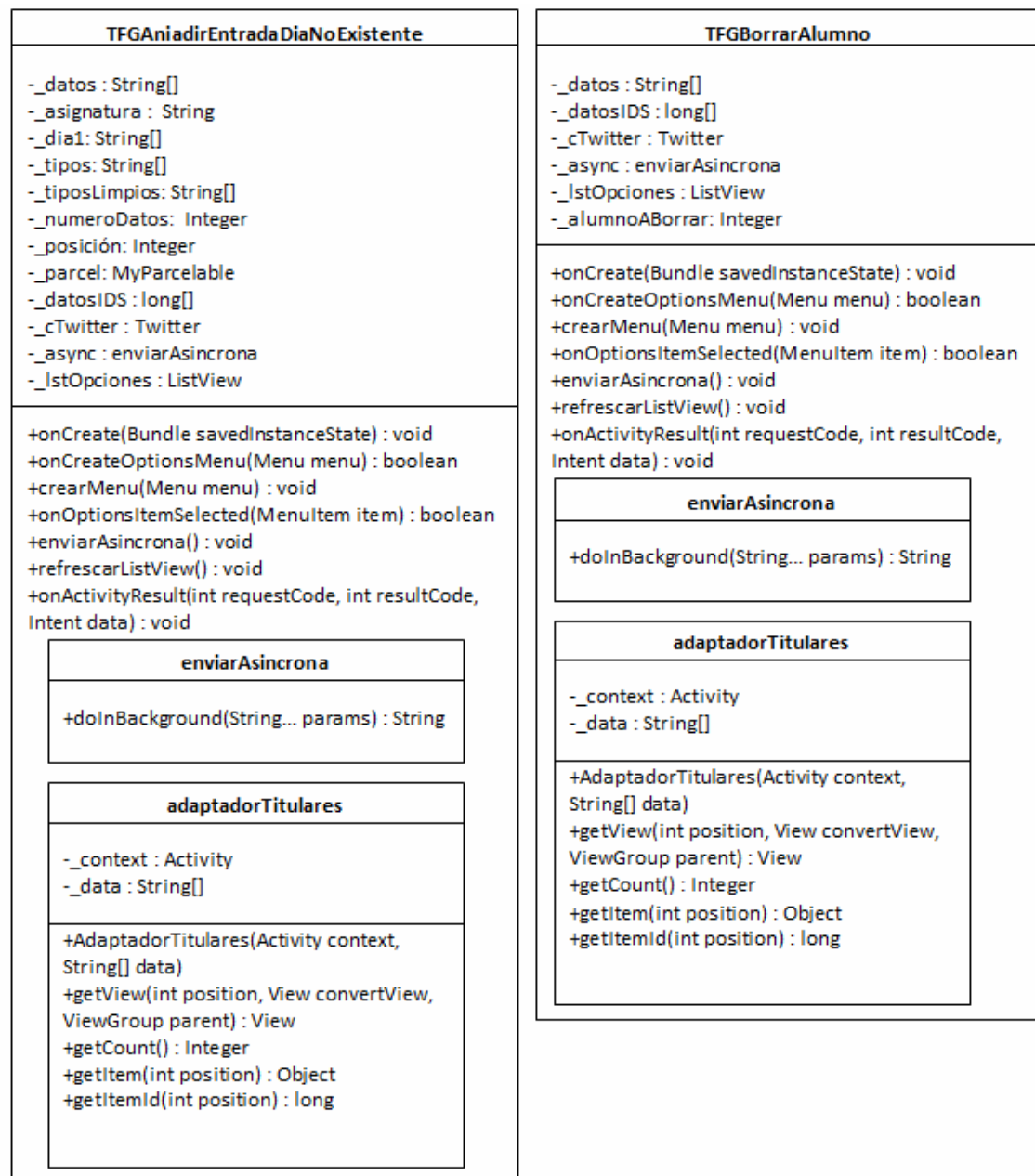


Figura 4.3.2.3: diagrama 3

En la figura 4.3.2.4 mostrada a continuación muestra las clases TFGHall y TFGModificarEntrada. Ambas clases son actividades y tienen como propósito el listado de los recursos del tablón y la modificación de los recursos existentes en el mismo respectivamente. Ambas clases poseen una clase dentro de sí, que crea y maneja un hilo de ejecución asíncrono encargado de las conexiones de red.

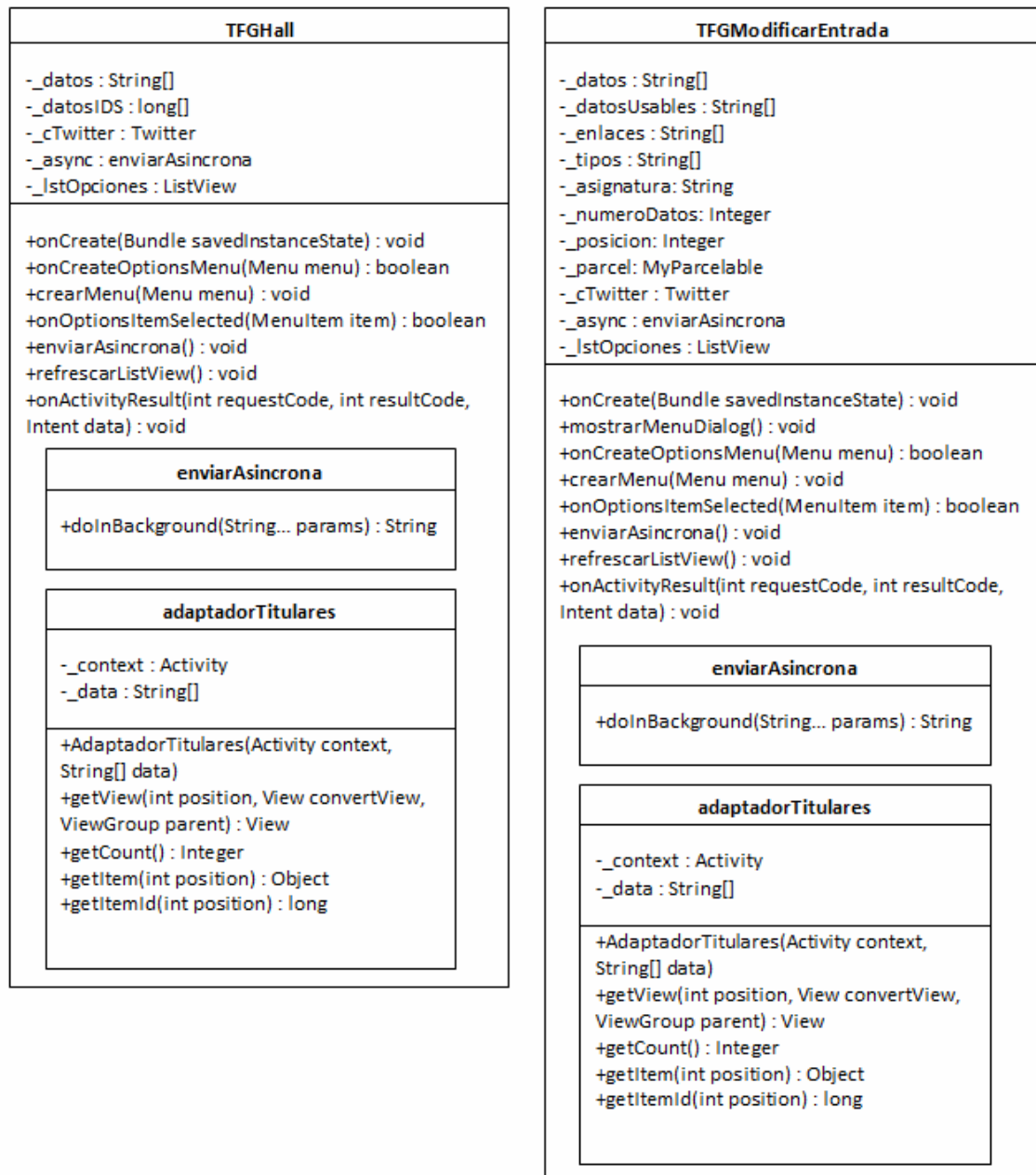


Figura 4.3.2.4: diagrama 4

En la figura 4.3.2.5 se muestran las clases `TFGModificarEntradaFormulario` y `TFGProfesActivity`. Ambas clases son actividades, y tienen como objetivo el mostrado de un formulario para la modificación de los datos correspondientes al recurso seleccionado, y la redirección a Twitter para realizar la identificación respectivamente. Ambas clases poseen una clase dentro de sí, que crea y maneja un hilo de ejecución asíncrono encargado de las conexiones de red.

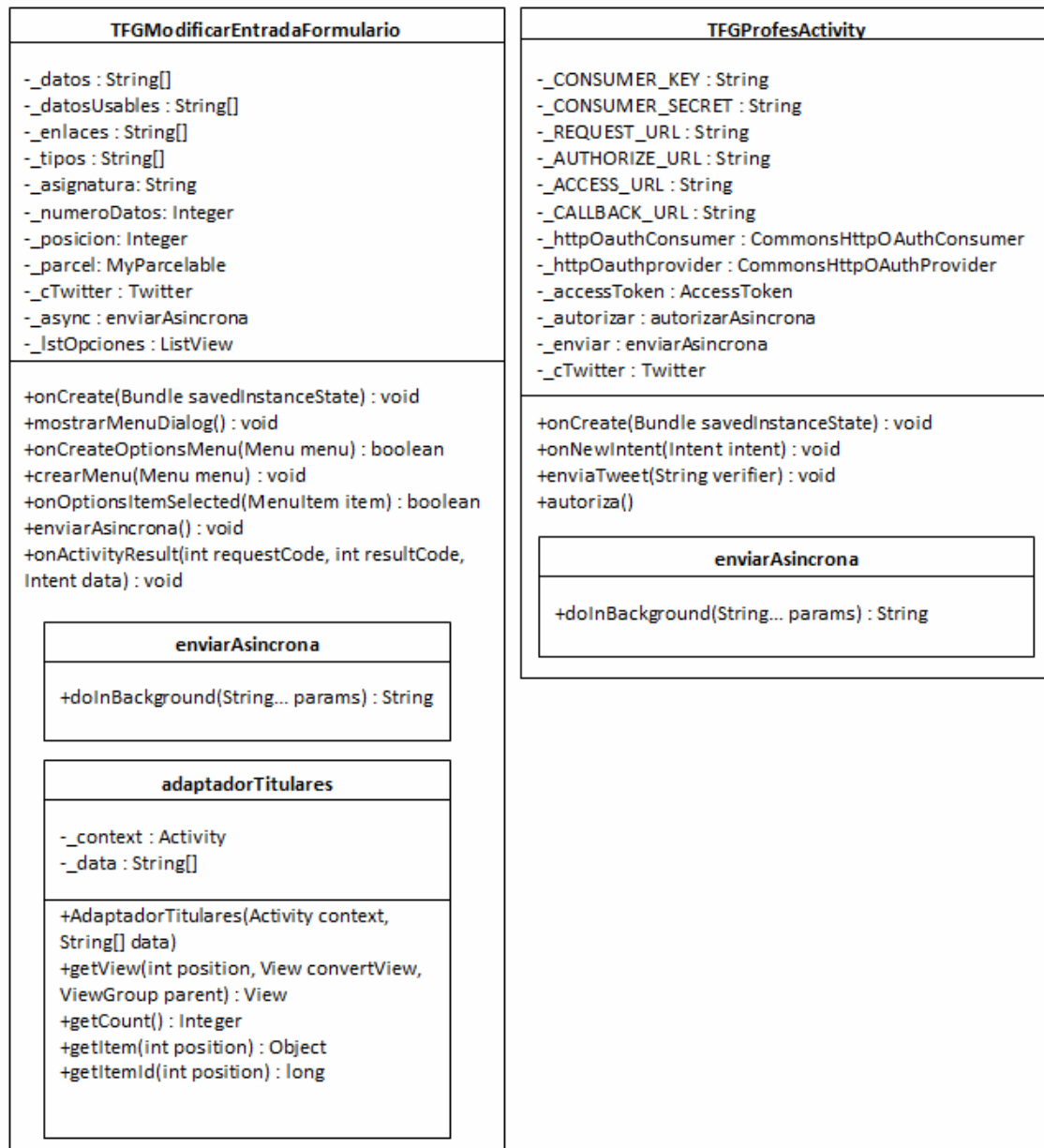


Figura 4.3.2.5: diagrama 5

En la figura 4.3.2.6 se muestran las clases TFGExploradorFicheros y ExploradorFTP. Ambas clases son actividades y tienen como propósito la exploración de ficheros localizados en el terminal móvil para su compartición y la exploración de ficheros ubicados en un servidor remoto, en este caso FTP, para su compartición respectivamente.

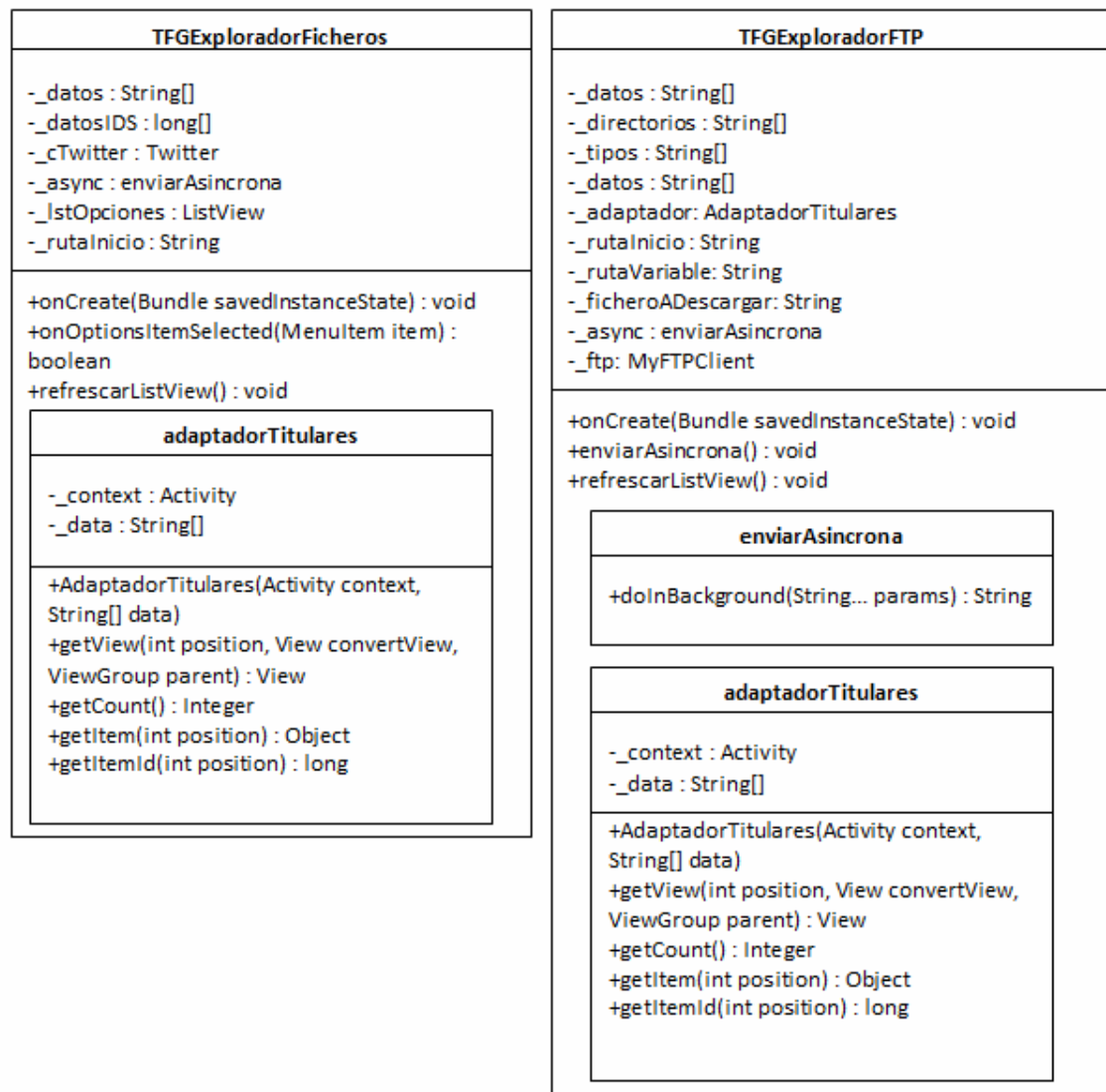


Figura 4.3.2.6: diagrama 6

4.4 DIAGRAMAS DE SECUENCIA

Una vez realizado el diagrama de clases perteneciente a la aplicación se definirán los diagramas de secuencia de la aplicación. Mediante estos se van a determinar las relaciones existentes entre los objetos que componen la aplicación para las principales funcionalidades que ofrecerá este software.

4.4.1 IDENTIFICACIÓN ANTE TWITTER

En la figura 4.4.1 que se ofrece a continuación se muestran las acciones realizadas por la aplicación al identificarse ante *Twitter*.

En este diagrama se muestra como un objeto de la clase TFGActivity, la cual actúa como actividad principal de la aplicación *MoLe Alumnos*, interactúa con *Twitter* para la identificación del usuario. Para ello se realizan las siguientes acciones:

1. La actividad TFGActivity redirige el flujo de ejecución al navegador Web, a la página de *Twitter*, para que el usuario se identifique ante el.
 - 1.1. Tras la identificación del usuario en el navegador la ejecución se redirige hacia la actividad TFGActivity nuevamente, recibiendo el *token* de sesión del usuario.
2. La actividad TFGActivity empaqueta el *token* para su envío al resto de actividades.

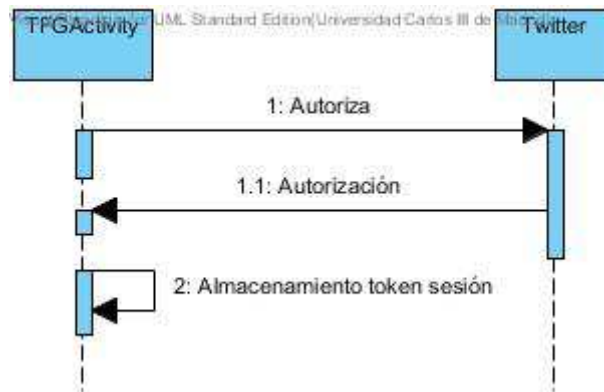


Figura 4.4.1: diagrama 1

La figura 4.4.1 representaría de igual modo el proceso de autenticación para la aplicación *MoLe* Profesores, siendo la actividad principal en este caso TFGProfesActivity, por lo que no será representada.

4.4.2 LISTADO DE CURSOS

En la figura 4.4.2 que se ofrece a continuación se muestra el listado de cursos suscritos por el alumno, dentro de la aplicación *MoLe* Alumnos, y las acciones llevadas a cabo por la actividad TFGHall para este fin. Para ello se realizan las siguientes acciones:

1. La actividad TFGHall recupera el objeto de sesión de *Twitter* que la actividad principal almacenó tras la autenticación del alumno ante *Twitter*.
2. La actividad TFGHall realiza una consulta a *Twitter* de las suscripciones que el alumno posee.
3. *Twitter* devuelve el listado con los identificadores numéricos de cada suscripción existente para el alumno.
4. La actividad TFGHall realiza una consulta a *Twitter* por cada identificador de curso recibido para conseguir el nombre del mismo.
5. *Twitter* devuelve los nombres pertenecientes a cada identificador de curso que TFGhall solicitó.
6. TFGHall muestra los cursos suscritos por pantalla, en forma de lista.

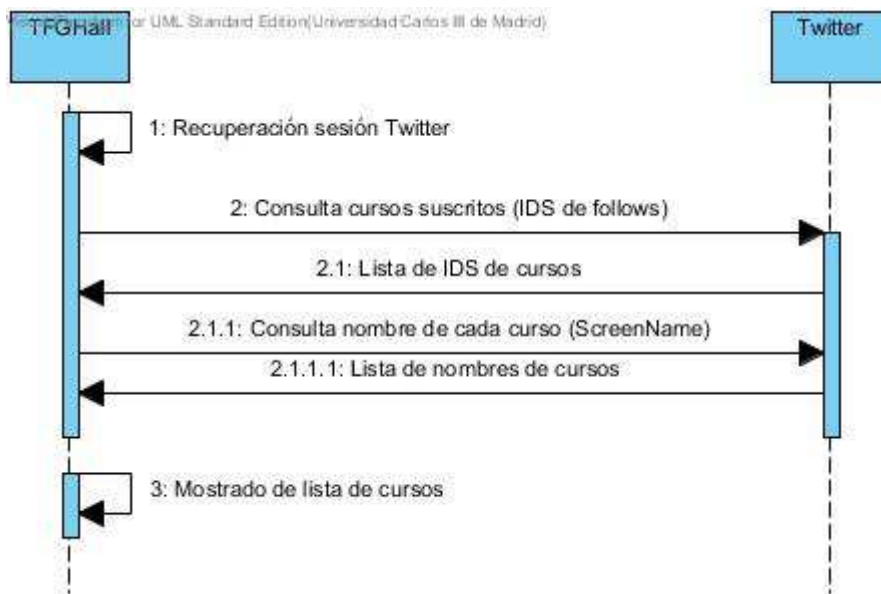


Figura 4.4.2: diagrama 2

4.4.3 BORRAR CURSO

La figura 4.4.3 que se ofrece a continuación muestra el proceso realizado por la actividad TFGBorrarCurso para borrar un curso suscrito por el alumno, dentro de la aplicación *MoLe Alumnos*. Para ello se realizarán las siguientes acciones:

1. La actividad TFGBorrarCurso recupera el objeto de sesión de *Twitter* que la actividad principal almacenó tras la autenticación del alumno ante *Twitter*.
2. La actividad TFGBorrarCurso realiza una consulta a *Twitter* de las suscripciones que el alumno posee.
3. *Twitter* devuelve el listado con los identificadores numéricos de cada suscripción existente para el alumno.
4. La actividad TFGBorrarCurso realiza una consulta a *Twitter* por cada identificador de curso recibido para conseguir el nombre del mismo.
5. *Twitter* devuelve los nombres pertenecientes a cada identificador de curso que TFGBorrarCurso solicitó.
6. TFGBorrarCurso muestra los cursos suscritos por pantalla, en forma de lista.
7. Del listado de cursos mostrado se selecciona uno de ellos, y se le envía a *Twitter* para su borrado.
8. *Twitter* informa del borrado correcto del curso deseado.
9. La actividad TFGBorrarCurso realiza una consulta a *Twitter* de las suscripciones que el alumno posee nuevamente.

10. *Twitter* devuelve el listado con los identificadores numéricos de cada suscripción existente para el alumno.
11. La actividad *TFGBorrarCurso* realiza una consulta a *Twitter* por cada identificador de curso recibido para conseguir el nombre del mismo.
12. *Twitter* devuelve los nombres pertenecientes a cada identificador de curso que *TFGBorrarCurso* solicitó.
13. *TFGBorrarCurso* muestra los cursos suscritos por pantalla, actualizando la lista anteriormente mostrada con las suscripciones actuales del alumno.

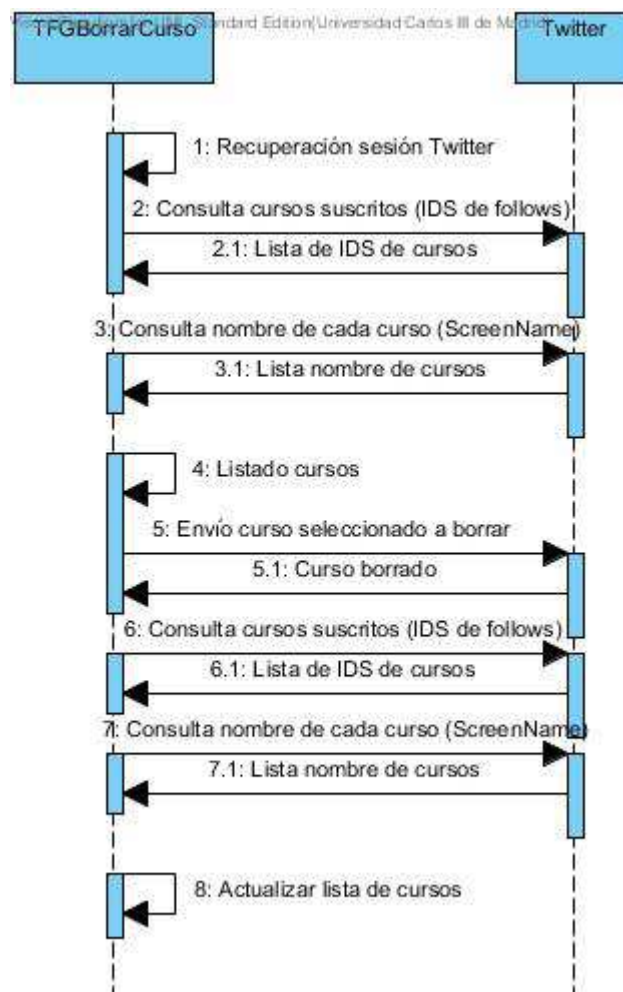


Figura 4.4.3: diagrama 3

4.4.4 MOSTRAR TABLÓN

La figura 4.4.4 que se ofrece a continuación muestra el proceso realizado por la actividad *TFGCurso* para mostrar los contenidos pertenecientes al tablón de un curso seleccionado. Para ello se realizarán las siguientes acciones:

1. La actividad TFGCurso recupera el objeto de sesión de *Twitter* que la actividad principal almacenó tras la autenticación del alumno ante *Twitter*.
2. La actividad TFGCurso recupera el nombre del curso del que se desea acceder a su tablón.
3. La actividad TFGCurso solicita al servidor el tablón perteneciente al curso, cuyo nombre se recuperó en el punto 2.
4. El servidor envía el fichero XML con el tablón del curso solicitado.
5. La actividad TFGCurso procesa el fichero XML del tablón y recupera los campos que este posee.
6. La actividad TFGCurso muestra el contenido del tablón.

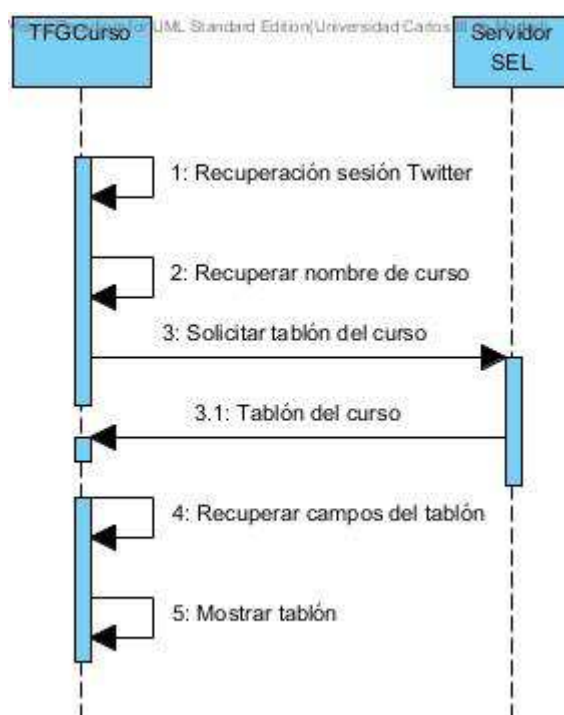


Figura 4.4.4: diagrama 4

La figura 4.4.4 representaría de igual modo el proceso para mostrar el tablón de la asignatura dentro de la aplicación *MoLe* Profesores, con la salvedad de que la actividad que realiza la comunicación frente al servidor es la actividad TFGHall de dicha aplicación, y el nombre del curso se recupera del propio identificador del usuario de *Twitter* identificado. Por esta razón no ha sido necesario incluir este diagrama.

4.4.5 DESCARGA DE MATERIAL

En la figura 4.4.5 que se ofrece a continuación se muestra como a partir del tablón mostrado en la figura 4.4.4 se puede seleccionar un recurso, contenido en el mismo, y realizar su descarga. Para ello se realizarán los siguientes pasos:

1. Se selecciona un recurso existente en el tablón en la actividad TFGCurso.
2. La actividad TFGCurso procesa la selección, recupera el nombre del fichero relacionado con la misma y realiza la petición de descarga del recurso al servidor.
3. El servidor envía el recurso seleccionado.

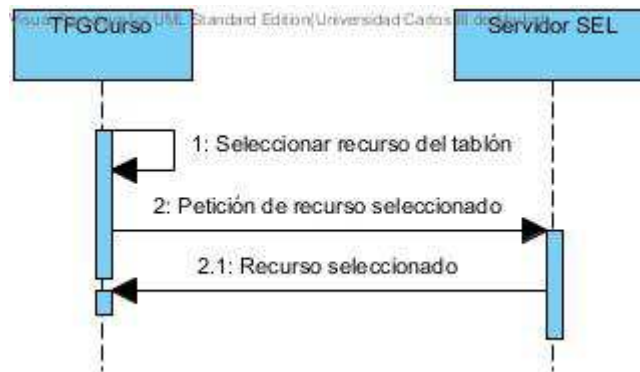


Figura 4.4.5: diagrama 5

4.4.6 SUSCRIBIR CURSO

La figura 4.4.6 que se ofrece a continuación muestra el proceso realizado por la actividad TFGAniadirCurso para suscribir al alumno a nuevos cursos. Para ello se realizarán los siguientes pasos:

1. La actividad TFGAniadirCurso recupera el objeto de sesión de *Twitter* que la actividad principal almacenó tras la autenticación del alumno ante *Twitter*.
2. La actividad TFGAniadirCurso enviará a *Twitter* el nombre del curso al que el alumno desea suscribirse.
3. *Twitter* responde a la actividad si el curso ha sido añadido correctamente, o no.



Figura 4.4.6: diagrama 6

4.4.7 LISTADO DE ALUMNOS

En la figura 4.4.7 que se ofrece a continuación se muestra el proceso por el cual la actividad TFGAlumnos genera una lista con los alumnos suscritos al curso. Para ello se realizarán los siguientes pasos:

1. La actividad TFGAlumnos recuperará el objeto de sesión de *Twitter* que la actividad principal almacenó tras la autenticación del profesor ante *Twitter*.
2. La actividad TFGAlumnos solicitará a *Twitter* una lista con los identificadores de los alumnos que se encuentran suscritos al curso.
3. *Twitter* enviará a la aplicación la lista solicitada.
4. La actividad TFGAlumnos solicitará a *Twitter* el nombre de cada alumno en base a su identificador anteriormente obtenido.
5. *Twitter* devolverá el nombre de todos los alumnos cuyo nombre ha sido solicitado en base a su identificador.
6. La actividad TFGAlumnos mostrará los alumnos suscritos al curso en forma de lista.

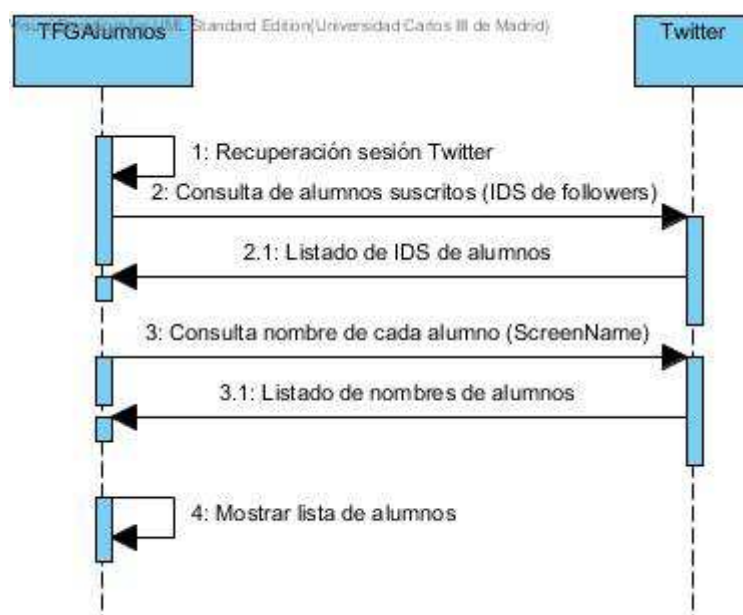


Figura 4.4.7: diagrama 7

4.4.8 BORRADO DE ALUMNOS

En la figura 4.4.8 que se ofrece a continuación se ofrece el proceso por el cual la actividad TFGBorrarAlumno procede al borrado de un alumno que se encuentra suscrito al curso. Para ello se realizan los siguientes pasos:

1. La actividad TFGBorrarAlumno recuperará el objeto de sesión de *Twitter* que la actividad principal almacenó tras la autenticación del profesor ante *Twitter*.
2. La actividad TFGBorrarAlumno solicitará a *Twitter* una lista con los identificadores de los alumnos que se encuentran suscritos al curso.
3. *Twitter* enviará a la aplicación la lista solicitada.
4. La actividad TFGBorrarAlumno solicitará a *Twitter* el nombre de cada alumno en base a su identificador anteriormente obtenido.
5. *Twitter* devolverá el nombre de todos los alumnos cuyo nombre ha sido solicitado en base a su identificador.
6. La actividad TFGBorrarAlumno mostrará los alumnos suscritos al curso en forma de lista.
7. La actividad TFGBorrarAlumno seleccionará un alumno y lo enviará a *Twitter* para su borrado.
8. *Twitter* confirmará el borrado del alumno.
9. La actividad TFGBorrarAlumno solicitará a *Twitter* una lista con los identificadores de los alumnos que se encuentran suscritos al curso.
10. *Twitter* enviará a la aplicación la lista solicitada.
11. La actividad TFGBorrarAlumno solicitará a *Twitter* el nombre de cada alumno en base a su identificador anteriormente obtenido.
12. *Twitter* devolverá el nombre de todos los alumnos cuyo nombre ha sido solicitado en base a su identificador.
13. La actividad TFGBorrarAlumno mostrará los alumnos suscritos al curso en forma de lista actualizados.

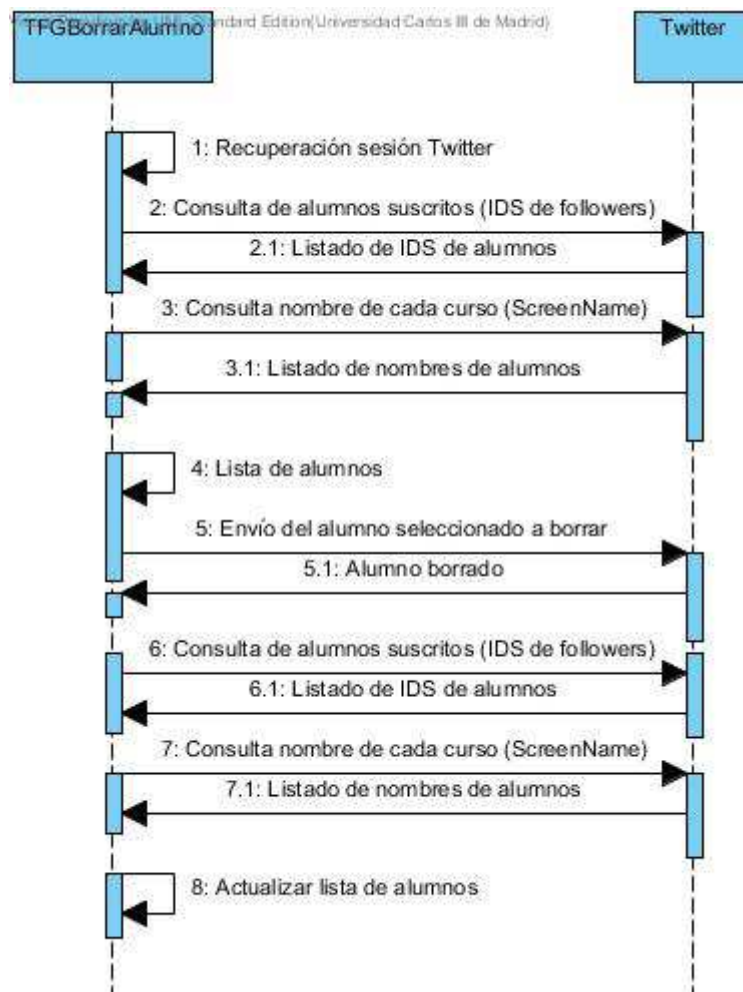


Figura 4.4.8: diagrama 8

4.4.9 AÑADIR CONTENIDO

La figura 4.4.9 ofrecida a continuación muestra como la actividad TFGAniadirEntrada se encarga de incluir nuevos contenidos en el tablón del curso, y los diferentes elementos con los que establece comunicación. Para ello se realizan los siguientes pasos:

1. La actividad TFGAniadirEntrada recupera los campos del tablón del curso.
2. La actividad TFGAniadirEntrada muestra el contenido del tablón del curso en forma de lista.
3. En la actividad TFGAniadirEntrada se decide incluir un nuevo recurso, por lo que se ejecuta un formulario para añadir la información de dicho recurso.
4. La actividad que posee el formulario recoge los campos rellenos para la creación del nuevo recurso.
5. La actividad que posee el formulario actualiza el contenido del tablón existente en memoria.

6. La actividad que posee el formulario constituye un fichero XML actualizado para dicho tablón, con los contenidos del tablón actualizados.
7. La actividad que posee el formulario envía al servidor el recurso que se desea compartir.
8. El servidor responde a dicha actividad del estado de la recepción del recurso.
9. La actividad que posee el formulario envía al servidor el fichero XML perteneciente a los contenidos compartidos en el tablón actualizado con el nuevo recurso incluido.
10. El servidor responde a dicha actividad del estado de la recepción del recurso.
11. La actividad que posee el formulario envía a TFGAniadirEntrada el tablón actualizado con los nuevos contenidos.
12. La actividad TFGAniadirEntrada recupera los campos pertenecientes al tablón.
13. La actividad TFGAniadirEntrada muestra el contenido del tablón actualizado.

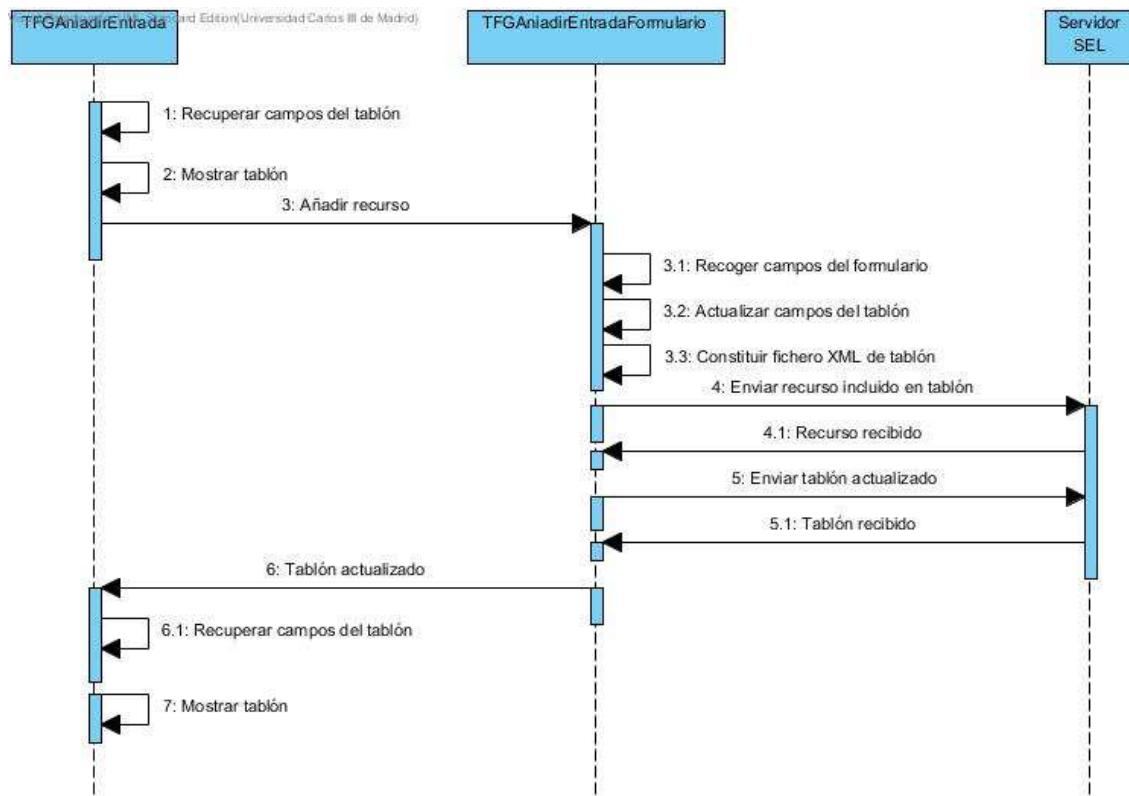


Figura 4.4.9: diagrama 9

4.4.10 BORRAR CONTENIDO

La figura 4.4.10 muestra el proceso por el cual la actividad TFGModificarEntrada borra un recurso existente en el tablón del curso. Para ello se realizan los siguientes pasos:

1. La actividad TFGModificarEntrada recupera los campos referentes al tablón del curso.
2. La actividad TFGModificarEntrada muestra el contenido del tablón en forma de lista.
3. El usuario selecciona de la lista de recursos del tablón cual quiere modificar.
4. La actividad TFGModificarEntrada borra el contenido referente a la entrada seleccionada.
5. La actividad TFGModificarEntrada actualiza los campos referentes al tablón que se encuentran en memoria.
6. La actividad TFGModificarEntrada constituye un fichero XML con el contenido del tablón actualizado tras el borrado del recurso.
7. La actividad TFGModificarEntrada envía el tablón del curso contenido en el fichero XML.
8. El servidor responde sobre el estado de la recepción del fichero.
9. La actividad TFGModificarEntrada muestra el contenido del tablón, actualizado tras el borrado del recurso seleccionado anteriormente.

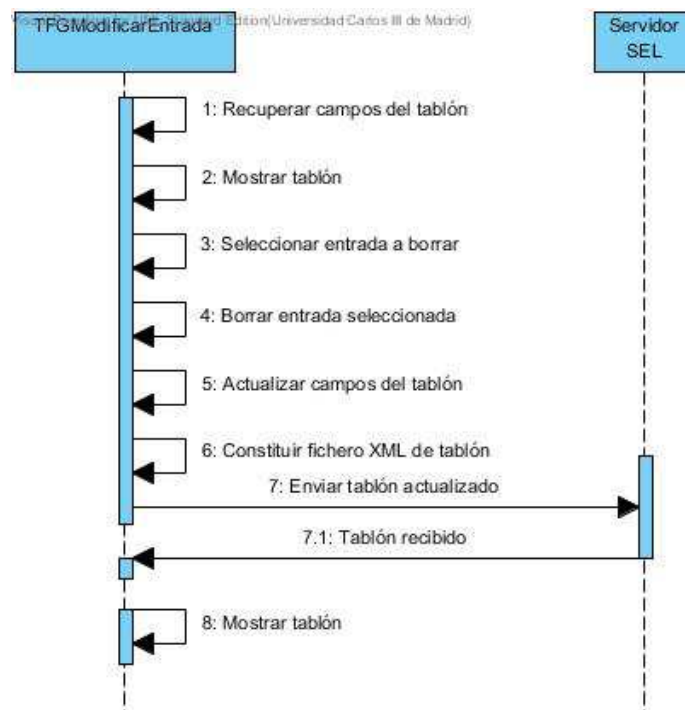


Figura 4.4.10: diagrama 10

4.4.11 MODIFICAR CONTENIDO

La figura 4.4.11 que se ofrece a continuación muestra el proceso por el cual la actividad TFGModificarEntrada realiza cambios sobre recursos existentes en el tablón del curso. Para ello se realizan los siguientes pasos:

1. La actividad TFGModificarEntrada recupera los campos del tablón del curso.
2. La actividad TFGModificarEntrada muestra el contenido del tablón del curso en forma de lista.
3. El usuario el recurso que desea modificar de la lista anteriormente mostrada, y envía los datos del recurso al formulario de modificación.
4. La actividad que posee el formulario muestra el contenido del recurso actual y permite la modificación de sus datos.
5. La actividad que posee el formulario actualiza el contenido del tablón existente en memoria.
6. La actividad que posee el formulario constituye un fichero XML actualizado para dicho tablón, con los contenidos del tablón actualizados.
7. La actividad que posee el formulario envía al servidor el recurso que se desea compartir.
8. El servidor responde a dicha actividad del estado de la recepción del recurso.
9. La actividad que posee el formulario envía al servidor el fichero XML perteneciente a los contenidos compartidos en el tablón actualizado con el nuevo recurso incluido.
10. El servidor responde a dicha actividad del estado de la recepción del recurso.
11. La actividad que posee el formulario envía a TFGModificarEntrada el tablón actualizado con los nuevos contenidos.
12. La actividad TFGModificarEntrada recupera los campos pertenecientes al tablón.
13. La actividad TFGModificarEntrada muestra el contenido del tablón actualizado.

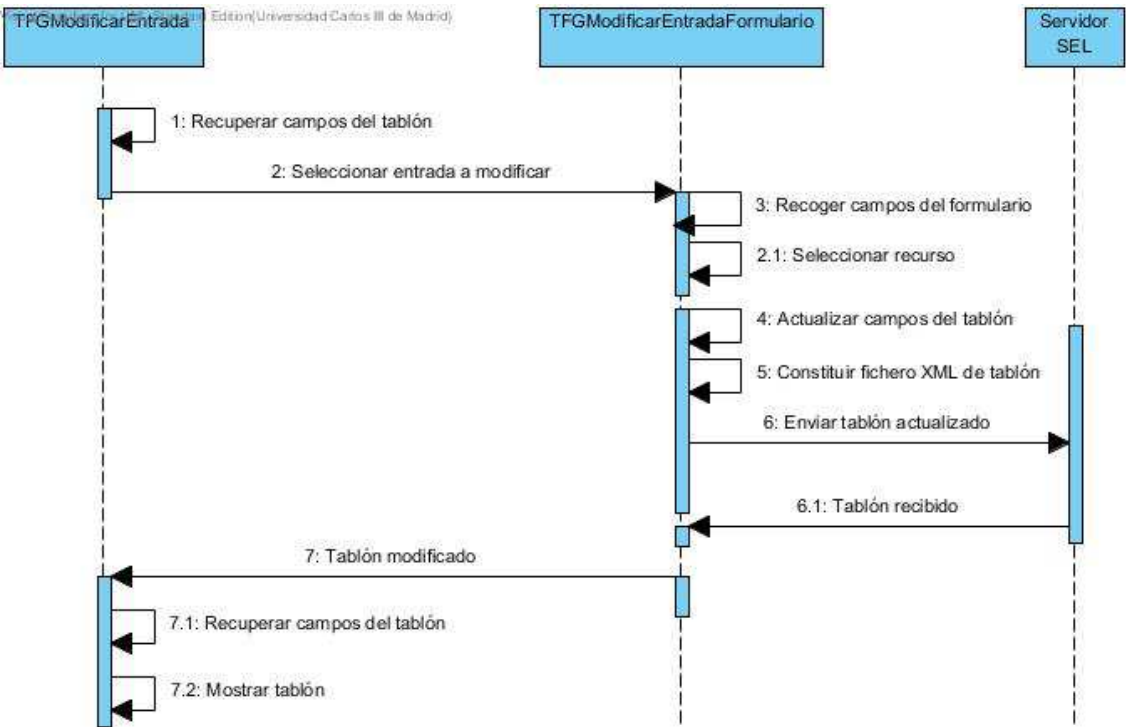


Figura 4.4.11: diagrama 11

CAPÍTULO 5 : PRUEBAS DE LA APLICACIÓN

5.1 PRUEBAS UNITARIAS	89
5.2 PRUEBAS DE INTEGRACIÓN	91
5.3 PRUEBAS DE SISTEMA	92

En este capítulo se definen los casos de prueba unitaria, integración y sistema realizados para determinar el correcto funcionamiento de la aplicación.

Las pruebas software son investigaciones empíricas y técnicas cuyo fin es proporcionar información objetiva e independiente sobre la calidad del producto desarrollado, siendo este hecho de especial interés para los *stakeholders*. Dichas actividades forman parte del proceso de control de calidad existente en el desarrollo software.

5.1 PRUEBAS UNITARIAS

Las pruebas unitarias tienen como objetivo probar el correcto funcionamiento de un módulo determinado del código fuente perteneciente a la aplicación. Esto sirve para asegurar que cada módulo funciona correctamente por separado. Posteriormente se realizarán pruebas complementarias a estas, para comprobar otros aspectos igual de importantes a los probados en esta sección.

Dado que la aplicación posee una gran mayoría de clases de con carácter Activity sólo se han realizado pruebas unitarias sobre las clases `parserXML` y `MyFTPClient`, siendo ambas los pilares sobre los que se sostiene el funcionamiento de la aplicación, y cuya funcionalidad desarrollada es utilizada por todas las actividades que componen la aplicación.

Dado el carácter de los métodos contenidos en dichas clases las pruebas unitarias han sido realizadas llamando a dichos métodos de forma manual, alterando su entrada, utilizando valores conocidos y esperando los cambios que dichos métodos producen en la memoria de la tableta.

Para la clase `MyFTPClient` los métodos sometidos a pruebas unitarias han sido los siguientes:

- **Conectar:** Método que establece comunicación con el servidor FTP remoto, y que retorna true o false en base a dicha conexión.
- **Ubicación:** Método que devuelve la ruta completa de la carpeta sobre la que se esta situado dentro del servidor FTP.
- **CambiarDir:** Método que recibe el nombre de una carpeta en formato cadena de caracteres y que realiza un cambio de directorio a dicha carpeta.
- **DescargarFich:** Método que recibe el nombre de un fichero en formato cadena de caracteres y que inicia la descarga de dicho recurso a la tableta.
- **SubirFich:** Método que recibe la ruta completa de un fichero alojado en memoria de la tableta, y que lo sube al servidor FTP.
- **BorrarFich:** Método que recibe la ruta completa de un fichero alojado en memoria de la tableta y que lo elimina.

- **CrearFich:** Método que recibe una ruta completa de un fichero nuevo y su contenido, creándolo en la memoria de la tableta.
- **ListarElementos:** Método que devuelve el listado de elementos contenidos dentro de una carpeta del servidor FTP.

En la tabla 5.1.1 se ofrecen un resumen de la batería de pruebas realizada sobre los métodos anteriormente mencionados, explicando los valores que ha recibido cada método, el resultado esperado y el resultado obtenido. Ha sido necesario ejecutar varias veces algunos métodos tales como “ubicación” para comprobar el correcto funcionamiento de otros cuyas acciones no dejan huella presente en el servidor y/o no devuelven valores.

Método	Entrada	Resultado esperado	Resultado obtenido	Estado
conectar	N/A	True	True	Correcto
Ubicación	N/A	/home/seladmin	/home/seladmin	Correcto
CambiarDir	dropbox	N/A	N/A	Correcto
Ubicación	N/A	/home/seladmin/dropbox	/home/seladmin/dropbox	Correcto
DescargarFich	Alg_t1.pdf	Fichero descargado	Fichero descargado	Correcto
SubirFich	/mnt/sdcard/TFG/pUnit.pdf	Fichero subido	Fichero subido	Correcto
BorrarFich	/mnt/sdcard/TFG/pUnit.pdf	Fichero borrado	Fichero borrado	Correcto
CrearFich	/mnt/sdcard/TFG/a.txt ; ZZ	Fichero creado	Fichero creado	Correcto
ListarElementos	N/A	Alg_t1.pdf, pUnit.pdf	Alg_t1.pdf, pUnit.pdf	Correcto

Tabla 5.1.1: Pruebas unitarias 1

Para la clase MyFTPClient los métodos sometidos a pruebas unitarias han sido los siguientes:

- **ProcesarFichero:** Método que procesa un fichero XML con estructura de tablón existente en la memoria de la tableta, y extrae los campos contenidos en el.
- **getRecursosDelTablon:** Método que devuelve los nombres de los recursos y días existentes en el tablón.
- **getTipoRecursosDelTablon:** Método que devuelve el tipo de dato para cada recurso existente en el tablón.
- **getEnlacesDelTablon:** Método que devuelve los enlaces pertenecientes a los recursos existentes en el tablón.

Para probar la funcionalidad descrita anteriormente se ha utilizado un fichero de prueba que poseía el siguiente contenido:

<s>

<h>Día 1</h>

Material 1Material1.pdf

<s>

<h>Día 2</h>

Material 2Material2.pdf

Tema 1Material3.pdf

Para la batería de pruebas realizada ha sido necesario introducir el fichero anteriormente mencionado en la memoria de la tableta, e instanciar un objeto de la clase parserXML enviando a su constructor la ruta del fichero de prueba creado.

En la tabla 5.1.2 que se ofrece a continuación se recogen los datos obtenidos de la batería de pruebas realizada sobre la clase parserXML indicando el método sobre el que se ha realizado la prueba, la entrada del mismo, la salida esperada, la salida recibida y el estado de la prueba.

Método	Entrada	Resultado esperado	Resultado obtenido	Estado
ProcesarFichero	N/A	N/A	N/A	-
getRecursosDelTablon	N/A	Día 1, Material 1, Día2, Material 2, Tema 1	Día 1, Material 1, Día 2, Material 2, Tema 1	Correcto
getTipoRecursoDelTablon	N/A	H, C, H, C, C	H, C, H, C, C	Correcto
getEnlacesDelTablon	N/A	Null, Material1.pdf, Null, Material2.pdf, Material 3.pdf	Null, Material1.pdf, Null, Material2.pdf, Material 3.pdf	Correcto

Tabla 5.1.2: Pruebas unitarias 2

5.2 PRUEBAS DE INTEGRACIÓN

La aplicación MoLe en su versión Android se comunica mediante un servidor que le provee de ficheros XML pertenecientes a los tabloneros y ficheros, en diferentes formatos, que contienen material docente. Al haberse hecho la integración desde el principio del desarrollo no se considera necesario realizar pruebas de integración en este sentido.

5.3 PRUEBAS DE SISTEMA

Dado que en todo el desarrollo del proyecto se han utilizado metodologías ágiles de desarrollo en las pruebas también se seguirán utilizando, de tal modo que el cliente sea el encargado de valorar si el proyecto cumple las expectativas que depositó en él. Para ello se deberán cumplir los siguientes puntos:

- El cliente deberá tomar parte activa en la creación de las pruebas de sistema.
- Únicamente se deben probar las cosas que se quiere que funcionen.

Para cada una de las pruebas de sistema deberán ser especificados una serie de campos, detallados a continuación, que recogen la información básica de dicha prueba de forma estructurada y simple. Para cada prueba se realizará una tabla que contendrá todos los campos que a continuación se detallan:

- **ID:** El identificador de la prueba esta compuesto por dos valores separados por un punto, siendo el primero referente a la funcionalidad probada en dicha prueba, y el segundo el número de prueba realizado sobre esa funcionalidad.
- **Descripción:** Define el objetivo que tiene marcado dicha prueba para que sea pasada.
- **Pasos:** Son la serie de pautas que se han de realizar en la aplicación para realizar la prueba.
- **Errores posibles:** Son los posibles errores que se han encontrado o pudieran encontrarse al realizar la prueba, y que no permiten que esta sea llevada a cabo.
- **Requisitos:** Son los requisitos con los que la prueba esta relacionada.
- **Estado:** Es el estado de la prueba tras su realización, pudiendo ser “pasada” si esta se ha completado con éxito, o “no pasada” si no ha sido completada correctamente.

A continuación en la Tabla 5.3.1 se muestra la plantilla que seguirán todas las pruebas de sistema realizadas:

ID	
Descripción	
Pasos	
Errores posibles	
Requisitos	
Estado	

Tabla 5.3.1: Plantilla 1

A continuación se van a especificar las pruebas realizadas para cada funcionalidad descrita en los diagrama de secuencia del capítulo anterior, además de funcionalidad extra que no quedó representada en dichos diagramas.

En la tabla 5.3.2 se muestra el resultado de la prueba realizada para comprobar el correcto arranque de la aplicación, es decir, que la aplicación sea capaz de iniciarse correctamente y que quede lista para su uso por parte del usuario.

ID	P1.01
Descripción	La aplicación deberá iniciarse correctamente
Pasos	1. Arrancar la aplicación
Errores posibles	Ninguno
Requisitos	RI-003
Estado	pasada

Tabla 5.3.2: Prueba sistema 1

En la tabla 5.3.3 se muestra el resultado de la prueba realizada para comprobar la identificación ante Twitter de la aplicación.

ID	P2.01
Descripción	Identificación del usuario
Pasos	<ol style="list-style-type: none">1. Arrancar la aplicación2. Pulsar el botón “Autenticar”
Errores posibles	<ul style="list-style-type: none">• El dispositivo no posee conexión a Internet.• El dispositivo esta conectado a una red Wifi que requiere inicio de sesión.• El servicio de Twitter está caído.
Requisitos	RI-003
Estado	pasada

Tabla 5.3.4: Prueba sistema 2

En la tabla 5.3.4 se muestra el resultado de la prueba realizada para comprobar el correcto listado de cursos suscritos por parte de un alumno.

ID	P3.01
Descripción	Listado de cursos
Pasos	<ol style="list-style-type: none">1. Arrancar la aplicación.2. Pulsar el botón “Autenticar”.3. Identificarse ante Twitter.
Errores posibles	<ul style="list-style-type: none">• El dispositivo no posee conexión a Internet.• El dispositivo esta conectado a una red Wifi que requiere inicio de sesión.• El servicio de Twitter está caído.
Requisitos	RI-004
Estado	pasada

Tabla 5.3.4: Prueba sistema 3

En la tabla 5.3.5 se muestra el resultado de la prueba realizada para comprobar el correcto funcionamiento del borrado de cursos suscritos por parte del alumno.

ID	P4.01
Descripción	Borrado de cursos
Pasos	<ol style="list-style-type: none">1. Arrancar la aplicación.2. Pulsar el botón “Autenticar”.3. Identificarse ante Twitter.4. Seleccionar el icono del “Lápiz”.5. Seleccionar un curso.6. Pulsar “Aceptar”
Errores posibles	<ul style="list-style-type: none">• El dispositivo no posee conexión a Internet.• El dispositivo está conectado a una red Wifi que requiere inicio de sesión.

	<ul style="list-style-type: none">• El servicio de Twitter está caído.
Requisitos	RI-007
Estado	pasada

Tabla 5.3.5: Prueba sistema 4

En la tabla 5.3.6 se muestra el resultado de la prueba referente a la suscripción a cursos nuevos por parte del alumno.

ID	P5.01
Descripción	Suscripción a curso
Pasos	<ol style="list-style-type: none">1. Arrancar la aplicación.2. Pulsar el botón “Autenticar”.3. Identificarse ante Twitter4. Seleccionar el icono de la “Lupa”.5. Escribir el curso al que se desea suscribir6. Pulsar “Añadir”
Errores posibles	<ul style="list-style-type: none">• El dispositivo no posee conexión a Internet.• El dispositivo esta conectado a una red Wifi que requiere inicio de sesión.• El servicio de Twitter está caído.• El curso no existe• Al nombre del curso no le precede el símbolo arroba (“@”)
Requisitos	RI-005
Estado	pasada

Tabla 5.3.6: Prueba sistema 5

En la tabla 5.3.7 se muestra el resultado de la prueba referente al mostrado del tablón de un curso suscrito

ID	P6.01
Descripción	Tablón curso
Pasos	<ol style="list-style-type: none">1. Arrancar la aplicación.2. Pulsar el botón “Autenticar”.3. Identificarse ante Twitter.4. Seleccionar un curso de los suscritos.
Errores posibles	<ul style="list-style-type: none">• El dispositivo no posee conexión a Internet.• El dispositivo está conectado a una red Wifi que requiere inicio de sesión.• El servicio de Twitter está caído.
Requisitos	RI-005
Estado	pasada

Tabla 5.3.7: Prueba sistema 6

En la tabla 5.3.8 se muestra el resultado de la prueba referente a la descarga de un material docente ofrecido en el tablón de un curso.

ID	P7.01
Descripción	Descargar recurso
Pasos	<ol style="list-style-type: none">1. Arrancar la aplicación.2. Pulsar el botón “Autenticar”.3. Identificarse ante Twitter.4. Seleccionar un curso de los suscritos.5. Seleccionar un recurso para su descarga.
Errores posibles	<ul style="list-style-type: none">• El dispositivo no posee conexión a Internet.• El dispositivo está conectado a una red Wifi que requiere inicio de sesión.

	<ul style="list-style-type: none">• El servicio de Twitter está caído.
Requisitos	RI-005
Estado	pasada

Tabla 5.3.8: Prueba sistema 7

En la tabla 5.3.9 se muestra el resultado de la prueba referente al listado de alumnos suscritos a un curso.

ID	P8.01
Descripción	Listado de alumnos
Pasos	<ol style="list-style-type: none">1. Arrancar la aplicación.2. Pulsar el botón “Autenticar”.3. Identificarse ante Twitter.4. Seleccionar el botón de los “hombrecitos”
Errores posibles	<ul style="list-style-type: none">• El dispositivo no posee conexión a Internet.• El dispositivo esta conectado a una red Wifi que requiere inicio de sesión.• El servicio de Twitter está caído.
Requisitos	RI-009
Estado	pasada

Tabla 5.3.9: Prueba sistema 8

En la tabla 5.3.10 se muestra el resultado de la prueba referente al borrado de alumnos suscritos a un curso.

ID	P9.01
Descripción	Borrado de alumnos
Pasos	<ol style="list-style-type: none">1. Arrancar la aplicación.2. Pulsar el botón “Autenticar”.3. Identificarse ante Twitter.4. Seleccionar el botón de los “hombrecitos”5. Seleccionar el botón del “Lápiz”6. Seleccionar el alumno a borrar.7. Pulsar el botón “Aceptar”
Errores posibles	<ul style="list-style-type: none">• El dispositivo no posee conexión a Internet.• El dispositivo esta conectado a una red Wifi que requiere inicio de sesión.• El servicio de Twitter está caído.
Requisitos	RI-010
Estado	pasada

Tabla 5.3.10: Prueba sistema 9

En la tabla 5.3.11 se muestra el resultado de la prueba referente al borrado de recursos ofrecidos en el tablón del curso.

ID	P10.01
Descripción	Borrado de contenidos
Pasos	<ol style="list-style-type: none">1. Arrancar la aplicación.2. Pulsar el botón “Autenticar”.3. Identificarse ante Twitter.4. Seleccionar el botón del “Lápiz”5. Seleccionar el recurso que se desea borrar.6. Pulsar “Borrar”
Errores posibles	<ul style="list-style-type: none">• El dispositivo no posee conexión a Internet.• El dispositivo esta conectado a una red Wifi que requiere inicio de sesión.• El servicio de Twitter está caído.
Requisitos	RI-011
Estado	pasada

Tabla 5.3.11: Prueba sistema 10

En la tabla 5.3.12 se muestra el resultado de la prueba referente a la modificación de recursos ofrecidos en el tablón del curso.

ID	P11.01
Descripción	Modificación de contenidos
Pasos	<ol style="list-style-type: none">1. Arrancar la aplicación.2. Pulsar el botón “Autenticar”.3. Identificarse ante Twitter.4. Seleccionar el botón del “Lápiz”5. Seleccionar el recurso que se desea borrar.6. Pulsar “Modificar”7. Rellenar el formulario de modificación.8. Pulsar el botón “Adjuntar”.9. Seleccionar el recurso a compartir.10. Pulsar el botón “modificar”
Errores posibles	<ul style="list-style-type: none">• El dispositivo no posee conexión a Internet.• El dispositivo esta conectado a una red Wifi que requiere inicio de sesión.• El servicio de Twitter está caído.
Requisitos	RI-011, RI-012, RI-013
Estado	pasada

Tabla 5.3.12: Prueba sistema 11

En la tabla 5.3.13 se muestra el resultado de la prueba referente a incluir nuevos recursos en el tablón del curso.

ID	P12.01
Descripción	Añadir contenido
Pasos	<ol style="list-style-type: none">1. Arrancar la aplicación.2. Pulsar el botón “Autenticar”.3. Identificarse ante Twitter.4. Seleccionar el botón del “Aspa”5. Seleccionar donde se desea añadir el recurso.6. Rellenar formulario.7. Pulsar botón “Adjuntar”8. Seleccionar recurso a compartir.9. Pulsar botón “Añadir”
Errores posibles	<ul style="list-style-type: none">• El dispositivo no posee conexión a Internet.• El dispositivo esta conectado a una red Wifi que requiere inicio de sesión.• El servicio de Twitter está caído.
Requisitos	RI-013, RI-014, RI-015
Estado	pasada

Tabla 5.3.13: Prueba sistema 12

CAPÍTULO 6 : CONCLUSIONES Y **LÍNEAS FUTURAS**

6.1 CONCLUSIONES	104
6.2 LÍNEAS FUTURAS	105
6.2.1 DIVERSIDAD DE FUENTES DE LOS RECURSOS	105
6.2.2 NUEVAS FUNCIONALIDADES	105
6.2.3 LIMITACIONES DE TWITTER4J	105
6.2.4 LIMITACIONES DE TWITTER.....	105

En este capítulo se incluyen las conclusiones obtenidas de la elaboración de este Trabajo de Fin de Grado, entre las que podemos diferenciar la evaluación personal de las experiencias ligadas al desarrollo de este trabajo y las líneas futuras para la aplicación, así como sus posibles mejoras.

6.1 CONCLUSIONES

Se ha desarrollado como Trabajo de Fin de Grado una aplicación móvil, en forma de cliente, que permite el aprendizaje en línea del usuario mediante el uso de dispositivos móviles Android, concretamente utilizando tabletas. La aplicación permite al alumno acceder a un aprendizaje en línea complementario a su estudios de carácter oficial, siendo este gestionado y dirigido por profesores, los cuales, puede, imparten los mismos cursos al alumno de forma presencial. La aplicación utiliza la identidad social del usuario para acceder a la misma, siendo Twitter la plataforma escogida por sus características favorables a las aplicaciones educativas y por su gran acogida a nivel mundial. El objeto de este trabajo ha sido el desarrollo de un módulo del cliente anteriormente descrito, el cual permite la compartición de material docente entre profesores y alumnos a través de la aplicación. Además este trabajo tenía como objetivo desarrollar una parte de un entorno de aprendizaje el cual no estuviera vinculado a ninguna plataforma, para lo cual se ha desarrollado una API de comunicación adaptable a cualquier tipo de plataforma con la que se desee trabajar.

Durante el desarrollo de este trabajo he puesto en práctica gran parte de los conocimientos adquiridos en estos cuatro años y medio de carrera, no obstante las vivencias relacionadas con las asignaturas vinculadas con el desarrollo de software han tenido especial importancia en el desarrollo de este trabajo. Además de para emplear los conocimientos adquiridos en las asignaturas cursadas en la carrera este trabajo me ha aportado conocimientos en gran variedad de aspectos que hasta el momento desconocía, tales como utilizar el API de Twitter, utilizar un servidor FTP desde Android, y una gran variedad de conocimientos técnicos relacionados con el desarrollo de aplicaciones Android, los cuales sería tedioso y poco práctico explicar aquí. Además este desarrollo ha supuesto para mí la elaboración de una aplicación desde cero, por lo que ha sido necesario utilizar una metodología de desarrollo de forma activa, cosa que hasta el momento nunca había seguido, al menos en un sentido estricto.

La realización de este Trabajo de Fin de Grado, a mi juicio, ha resultado interesante, y especialmente útil en los tiempos corren, dado que los terminales móviles forman parte de la vida cotidiana de una gran mayoría de ciudadanos, por lo que los convierten en un foco de desarrollo e innovación que está siendo actualmente explotado, pero sobre el que aun queda mucho por decir y mejorar. Por ello los conocimientos adquiridos en este trabajo son especialmente importantes.

En el desarrollo de este trabajo han surgido numerosas dificultades, relacionadas con la funcionalidad a desarrollar, las cuales han sido corregidas o controladas empleando gran cantidad de tiempo y esfuerzo. Entre estas dificultades podemos destacar la problemática asociada a las restricciones inherentes a Twitter, con la cual se ha tenido que aprender a convivir, amoldando el desarrollo a lo que este permitía, o la problemática asociada a utilizar tareas asíncronas, que tenían que operar bajo el mando de una actividad, y en ocasiones concretas parecían no querer seguir órdenes.

La duración del proyecto se estableció que sería de 300 horas, dividida en unos 4 meses, tiempo comprendido desde Octubre hasta Enero, ambos incluidos. La realidad es que el desarrollo de la aplicación ha requerido mucho más tiempo del que inicialmente se preveía iba a necesitar, y la elaboración del presente documento añadió muchas más horas a la cantidad real de trabajo aplicado a este trabajo, habiendo utilizado unas 550 horas de trabajo para completarlo.

En lo referente a los objetivos que se establecieron para este Trabajo de Fin de Grado continuación se va a evaluar el estado actual de cada uno de ellos, explicando si se ha completado cada uno de ellos, o de no haber sido así el estado en el que actualmente se encuentran:

- **Crear una parte de un PLE que permita a los alumnos de estudios de carácter oficial seguir sus clases matriculadas de forma virtual, y de una forma complementaria, a las plataformas ya existentes:** Este era el objetivo principal de este trabajo, y se ha cumplido para la funcionalidad que había que desarrollar. Para completar este objetivo ha sido necesario completar los objetivos secundarios que se explican a continuación, y que eran indispensables para consumir este objetivo.
- **Crear una parte de un PLE que utilice la identidad social del usuario de otras plataformas:** Este objetivo tenía como consecuencia que el usuario pudiera identificarse en la aplicación utilizando la identidad social que el poseía en otra plataforma, en este caso Twitter, lo cual ha sido completado con éxito en base a las exigencias descritas en este documento.
- **Crear una parte de un PLE con un LMS variable:** Este objetivo tenía como objetivo dotar de flexibilidad al cliente del entorno de aprendizaje, para permitir que pudiera coger recursos de cualquier LMS. Lo cierto es que actualmente solo recibe contenidos de un servidor FTP que hemos utilizado para el desarrollo, pero se ha desarrollado una API de comunicación que puede ser usada para la interconexión del cliente con cualquier LMS, por esta razón el objetivo ha sido completado.

6.2 LÍNEAS FUTURAS

En esta sección se van a tratar las líneas futuras esperadas para este trabajo, indicando las facetas que podrían ser mejorados para su correcta implantación en el mercado comercial, así como funcionalidades que la aplicación actualmente no desarrolla. Estos puntos son independientes de cualquier objetivo tratado en este documento y no se han realizado porque pertenecen a otros módulos de la misma aplicación que serán asignados a otros alumnos cuyo Trabajo de Fin de Grado consistirá en continuar la labor que yo empecé.

6.2.1 DIVERSIDAD DE FUENTES DE LOS RECURSOS

Sería interesante mejorar la funcionalidad existente implementando el API de interconexión con el servidor de recursos para comunicarse con más sistemas, y de forma transparente de cara al usuario acceder a través del cliente del entorno de aprendizaje a los contenidos docentes, estén donde quiera que fuere alojado. Esta mejora dotaría al profesor de mayor flexibilidad, ya que no le obligaría a utilizar un proveedor de contenidos determinado, aumentando el volumen de contenidos compartidos en el cliente, y por lo tanto enriqueciendo la aplicación de cara al alumno.

6.2.2 NUEVAS FUNCIONALIDADES

Dado que la aplicación tiene una funcionalidad muy limitada sería conveniente añadir nuevos aspectos funcionales que la hicieran más atractiva, tales como realización de autotests, tabloneros de preguntas por curso, mensajería entre usuarios, ya sea de alumno a alumno o de alumno a profesor, o viceversa, y por qué no, entrega de prácticas.

Todas y cada una de las funcionalidades descritas, de forma individual, darían para un Trabajo de Fin de Grado, por lo que en un futuro los alumnos que se interesen por este trabajo las realizarán.

6.2.3 LIMITACIONES DE TWITTER4J

Uno de los principales problemas existentes en el núcleo de la aplicación reside en la funcionalidad existente en las librerías de twitter4j del API de Twitter, las cuales no desarrollan toda la funcionalidad que el API ofrece, siendo en algunos aspectos dicha funcionalidad muy limitada o inexistente. Por ello sería conveniente desarrollar dicha funcionalidad para dotar a la aplicación de mayor dinamismo a la hora de utilizar los recursos existentes en Twitter.

6.2.4 LIMITACIONES DE TWITTER

El propio Twitter posee limitaciones de cara a los usuarios, para evitar que estos saturen el sistema con información no útil, y puedan resentir el rendimiento de sus servicios. Por ello Twitter establece límites a las consultas que un usuario puede realizar por hora, siendo este factor inapreciable cuando utilizas Twitter de forma convencional, pero pudiendo convertirse en un problema para este trabajo si en un curso hay demasiados alumnos. Por ello sería conveniente tratar de llegar a un acuerdo con Twitter, para intentar extender la limitación de consultas ligeramente, y por otro lado optimizar las consultas realizadas desde el propio cliente, economizando lo máximo posible cada una para que dicha restricción nunca suponga un problema.

BIBLIOGRAFÍA

- [1] SLIDESHARE. PLE, Personal Learning Environment [en línea].
<<http://www.slideshare.net/dreig/ple-1340811>> [Consulta: 10 Noviembre 2012].
- [2] SAMOO. LMS, Learning Management System [en línea].
<<http://samoo.es/index.php/es/component/content/article/36-master-your-menu>> [Consulta: 10 Noviembre 2012].
- [3] SLIDESHARE. Diseño de entornos de aprendizaje [en línea].
<<http://www.slideshare.net/mbulas/diseo-de-ambientes-de-aprendizaje>> [Consulta: 10 Noviembre 2012].
- [4] PLUSESMAS. Qué es y cómo funciona Facebook [en línea].
<http://www.plusesmas.com/nuevas_tecnologias/articulos/internet_email/que_es_y_como_funciona_facebook/123.html> [Consulta: 12 Noviembre 2012].
- [5] CIUDADANO2CERO. Qué es y cómo funciona Twitter [en línea].
<<http://www.ciudadano2cero.com/twitter-que-es-como-functiona/>> [Consulta: 13 Noviembre 2012].
- [6] SLIDESHARE. Introducción a las metodologías ágiles [en línea].
<<http://www.slideshare.net/Condiminds/introduccion-a-las-metodologas-giles>> [Consulta: 17 Noviembre 2012].
- [7] DOOLPHY. Aplicación Web. <<http://www.doolphy.com/es/>>
- [8] SOFTENG. Ventajas de utilizar Scrum [en línea]. <<http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>> [Consulta: 28 Noviembre 2012]
- [8] WC3SCHOOLS. Aplicación Android, 2012. <http://www.w3schools.com/>
- [9] BUSUU. Aplicación Android, 2012.
<https://play.google.com/store/apps/details?id=com.busuu.android.enc&hl=es>
- [10] BLACKBOARD. Aplicación Android, 2012.
https://play.google.com/store/apps/details?id=com.blackboard.android&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5ibGFja2JvYXJkLmFuZHIjaWQIXQ..
- [11] KHAN ACADEMY. Aplicación Android, 2012.
https://play.google.com/store/apps/details?id=home.widgetlabs&feature=search_result#?t=W251bGwsMSwxLDEsImhvbnVud2lkZ2V0bGFicyJd
- [12] WIKILEARNING. Aplicación Web, 2012. <http://www.wikilearning.com/>
- [13] SLIDESHARE. Aplicación Web, 2012. <<http://www.slideshare.net/>>

- [14] ROYAL PINGDOM. Cuota de mercado OS [en línea].
<http://royal.pingdom.com/2012/02/22/nokia-and-symbian-still-on-top-of-the-mobile-web-but-for-how-long/> Consulta: 28 Noviembre 2012]
- [15] LABIBLIADELPROGRAMADOR. Estructura Android [en línea].
<<http://labibliadelprogramador.blogspot.com.es/2012/09/estructura-de-android.html>>
Consulta: 28 Noviembre 2012]
- [16] SOFTENG. Ventajas de Scrum [en línea]. <<http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>> [Consulta: 28 Noviembre 2012]
- [17] UNIVERSIDAD CARLOS III DE MADRID. Revisión de Métrica Versión 3 [en línea]. <<http://es.scribd.com/doc/41708418/Metrica-3>> [Consulta: 29 Noviembre 2012]

APÉNDICES

APÉNDICE A: PRESUPUESTO DEL **PROYECTO**

Una vez conocido el esfuerzo necesario para la realización de este Trabajo de Fin de Grado se procederá a detallar los costes económicos asociados al mismo. En primer lugar se realizará un cómputo con los gastos asociados al esfuerzo humano, representado por las horas de trabajo empleadas por los miembros que han participado en el desarrollo, estableciendo un valor a cada hora de trabajo por cada miembro en base a su puesto dentro del propio desarrollo.

Como personal se ha incluido a Santiago García Delgado, como principal desarrollador de la aplicación, a Javier García Guzmán, el cual fue el tutor de este Trabajo y aportó horas al mismo en forma de reuniones y asesoramiento, y por último se ha incluido a Javier Sánchez Hernández, el cual ha aportado consejo y ayuda en el desarrollo de la aplicación, principalmente en la fase inicial del mismo.

Personal	Categoría	Coste hombre/hora	Duración (horas)	Coste (Euros)
Santiago García Delgado	Ingeniero Junior	12	400	4.800
Javier García Guzmán	Ingeniero Advance	50	34	1.700
Javier Sánchez Hernández	Ingeniero Senior	25	20	500
Hombre hora		87	Total	7.000

Tabla 7.1: Gastos 1

Además de los gastos relacionados con el esfuerzo humano, cuantificado anteriormente, es necesario representar el gasto asociado al material empleado para la realización de este proyecto. Dado que la duración del mismo ha sido de cuatro meses se realizará la amortización de dichos bienes con respecto a su tiempo real de uso en este Trabajo, y se realizará el cálculo de gastos en este ámbito acorde al tiempo de vida útil asociado a cada elemento.

Para el cálculo de las amortizaciones se ha considerado que, por un lado, la vida útil de los ordenadores portátiles y sobremesas es de 4 años, y que la vida útil de las tabletas es de 3 años.

El coste imputable por cada artículo ha sido calculado atendiendo a la siguiente fórmula:

$$\frac{\text{Coste del producto (€)} * \text{Tiempo de uso (Meses)} * \text{Utilización (0.0-1.0)}}{\text{Periodo de vida útil (Meses)}}$$

Periodo de vida útil (Meses)

Descripción	Coste (Euros)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable
Ordenador Toshiba Personal Computer, Procesador 2.27 GHz Intel i5, Memoria 4GB 1.066 DDR	700	100	1	48	14,6
Ordenador sobremesa, Procesador 2.67 GHz Intel i5 750, Memoria 4GB 1.600 DDR	700	100	4	48	58,4
Nexos 7 32GB	250	100	1	36	7
				Total	80

Tabla 7.2: Gastos 2

A continuación se muestra el coste total asociado a la elaboración de este Trabajo, sumando todos los costes existentes y calculados anteriormente. De este modo el coste del Trabajo asciende a 7.080 Euros (Siete mil ochenta Euros)

Presupuesto	Coste (Euros)
Coste personal	7.000
Coste amortizaciones	80
Coste Total	7.080

Tabla 7.3: Gastos 3